The 9825A uses HPL, a high level programming language, designed especially for scientists and engineers. HPL offers power and efficiency for handling equations, data manipulation, and input/output operations, yet it is easy to learn and use. This Quick Reference Guide summarizes all the HPL mnemonics associated with the 9825 and its optional Read Only Memories (ROMs).

Hewlett-Packard 9825A Calculator

Quick Reference Guide

HEWLETT-PACKARD CALCULATOR PRODUCTS DIVISION
P.O. Box 301, Loveland, Colorado 80537, Tel. (303) 667 5000

TABLE OF CONTENTS

GENERAL INFORMATION

The programming information in this booklet serves as a handy reference for users of the Hewlett-Packard 9825A Calculator. Use it as a pocket reference or keep it in a place near your calculator.

Turn-On

To turn the 9825A Calculator ON, set the power switch on the right-hand side of the calculator to the "1" position.

Available Read/Write Memory

Standard	6844 bytes	Opt. 002	23228 bytes
Opt. 001	15036 bytes	Opt. 003	31420 bytes

Range and Accuracy

The storage range of the 9825A is $-9.99999999999 \times 10^{99}$ through -1×10^{-99}, 0, 1×10^{-99} through $9.99999999999 \times 10^{99}$. The calculating range is $-9.99999999999 \times 10^{511}$ through -1×10^{-511}, 0, and 1×10^{-511} through $9.99999999999 \times 10^{511}$.
All numbers are represented internally with 12 significant digits accuracy.

Display and Line Length

The 9825A has a 5 x 7 dot matrix, 32-character LED display. Even though only 32 characters can be seen at one time, up to 80 characters can be keyed into the display. After 67 characters are entered, the calculator beeps to indicate approaching end of line. Up to 73 characters can be stored in a program line.

Character Set

The 16-character thermal strip printer and the 32-character LED display provide the following upper and lower case character set.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789:;<=>?!"#$%&' ()*+,-./↑[]⌈π_@

With the String or General I/O ROMs, the following characters are available:

ÅåÄäÖöÜüÆæÑñ¿ αβΓδΔθλμσΣτΦΩ X̄²£¶|→←↓⊢◀▒

Mathematical Hierarchy

highest priority	functions, flag references, r-variables
↑	↑ (exponentiation)
	implied multiplication
	unary minus
	* / mod
	+ −
	relational operators (=,>,<,<=,>=,#)
	not
	and
lowest priority	or xor

Errors

When an error occurs, the calculator beeps and an error number appears in the display.

SPECIAL KEYS

Control Keys

[CLEAR] Clears the display.

[CONTINUE] Automatically continues a program from where it was stopped.

[EXECUTE] Executes the single or multi-statement line which is in the display.

[RUN] Runs the program beginning at line 0.

[STOP] Stops the program at the completion of the currently executing line.

[STORE] Stores an individual program line or special function key definition in memory.

[SHIFT] When used with an alphanumeric key, the shifted form of the character or symbol is typed. The SHIFT key also releases shift lock.

[SHIFT LOCK] Locks the keyboard into the shifted state so that multiple shifted characters or symbols can be typed. The small light above the key indicates that shift lock is in effect.

Line Editing Keys

[DELETE] Deletes the program line in the display from the program in memory.

[INSERT] Inserts a line into a program before the fetched line when used in place of the STORE key.

[RECALL] Recalls the two previous entries back into the display. Press once for the most recent keyboard entry, and twice for the previous entry.

[FETCH] This typing aid key is used with the fetch command.

Character Editing Keys

[BACK] Moves the replace/insert cursor backwards in the display.

[FWD] Moves the replace/insert cursor forward in the display.

[DELETE] Deletes the character under the replace/insert cursor.

[INS RPL] Changes the replace cursor (▒) to an insert (◀) cursor and vice versa. When the replace cursor is present, each character typed replaces the character under the cursor. When the insert cursor is present, each character typed is inserted before the character under the cursor.

Special Function Keys

To define a special function key:

Press FETCH
Press fn, the special function key
Enter the key definition in the display
Press STORE

- Immediate Execute Keys are created when the contents of the key begins with an asterisk (*).
- Immediate Continue Keys are created when the contents of the key begins with a slash (/).
- Typing Aid Keys are created when the contents of the key begins with a character other than the asterisk or the slash.

System Command Keys

RESET — Returns the calculator and I/O cards to the power-on state without erasing programs or variables.

PRT ALL — Sets the print-all mode on or off. When "on", all executed lines, stored lines, messages and commands which are normally displayed, are also printed.

REWIND — Automatically rewinds the tape cartridge to its beginning.

STEP — Steps through the program executing one line at a time.

ERASE — This typing aid is used with the erase command.

LOAD — This typing aid is used with the load file (ldf) statement.

RECORD — This typing aid is used with the record file (rcf) statement.

LIST — This typing aid is used with the list statement.

OPERATORS

Assignment		
→	5 →A	Assigns values to variables
Arithmetic		
+	A+B	Add
−	X−2	Subtract
*	A*B	Multiply
	AB	Implied multiply
/	T/6	Divide
↑	2↑8	Exponentiate
mod	Amod4	Modulus

Relational		
=	P=Q	Equal to
>	X=Y	Greater than
<	S<T	Less than
>= or =>	B>=C	Greater than or equal to
<= or =<	C<=D	Less than or equal to
# or >< or <>	M#N	Not equal to

Logical
and
or
xor
not

Truth Table:

A	B	A and B	A or B	A xor B	not A
0	0	0	0	0	1
0	T	0	1	1	1
T	0	0	1	1	0
T	T	1	1	0	0

T=Any non-zero value or 1=True
0=False

SPECIAL FLAGS

The following flags are controlled automatically as described when special conditions are met.

Flag 12 — Automatically set to 1 when Extended I/O data transfer (tfr) is completed.

Flag 13 — Automatically set to 1 if no data is entered at an ent or enp statement, and the CONTINUE or STOP key is pressed. Automatically cleared to 0 if data is entered at an ent or enp statement.

Flag 14 — When set to 1 by the user, math errors (e.g., square root of a negative number) do not cause termination of the program.

Flag 15 — Automatically set to 1 when a math error occurs.

COMMANDS

Commands are operations which cannot be stored as part of a program. Each is followed by EXECUTE unless otherwise noted.

cont	cont	Continues program execution from current position in memory. This command is immediately executed when the CONTINUE key is pressed.
	cont 7	Continues program execution from the specified line number.
	cont "loop"	Continues program execution from the specified label.
del	del 9	Deletes a single program line.
	del 2,20	Deletes lines of program between and including the specified lines.
	del 1,15,*	Deletes lines of program between and including the specified lines. If the deleted lines are referenced in the remaining program lines, references to deleted lines are changed to the next line following the deleted lines.
erase	erase	Erases programs and variables from memory.
	erase a	Erases the entire calculator memory.
	erase v	Erases variables and flags from memory.
	erase k	Erases all special function keys from memory.
	erase fn	Erases the special function key from memory. This command is executed immediately when the special function key is pressed.
fetch	fetch	Fetches line 0 of the program into the display.
	fetch 8	Fetches the specified program line into the display.
	fetch fn	Fetches the special function key into the display. This command is executed immediately when the special function key is pressed.

list	list fn	Lists the special function key. This command is executed immediately when the special function key is pressed.
run	run	Runs the program in memory from the beginning. This command is immediately executed when the RUN key is pressed.
	run 15	Runs the program beginning at the specified line.
	run "update"	Runs the program beginning at the specified label.

STATEMENTS

beep	beep	Causes the calculator to output an audible beep.
cfg	cfg	Clears all flags (0 through 15) to 0.
	cfg 3,8,T	Clears the specified flags to 0.
cmf	cmf	Complements all flags (0 through 15). If flag = 0, it is set to 1; if flag = 1, it is cleared to 0.
	cmf A,B	Complements specified flags.
csv	csv	Clears simple variables A through Z to 0.
deg	deg	Specified degrees will be used for all calculations involving angles.
dim	dim S[1,6],T,I,[J] dim A[−5:15,0:3]	Reserves space for simple variables and arrays having the specified dimensions.
dsp	dsp "Sum=",S	Displays values or text on the calculator display.
	dsp "He said""NO""!"	Quote marks can be displayed within text if they are doubles.
end	end	Causes the program to stop and resets the program line counter to 0.
enp	enp P	Causes the program to wait for values to be entered from the keyboard. The values are printed and displayed as entered.
	enp "Price?",P	Prompts the user to enter values from the keyboard. Prompts and entered values are printed and displayed as entered.
ent	ent V	Causes the program to wait for values of variables to be entered from the keyboard.
	ent "Value",V	Prompts the user to enter values of variables from the keyboard.
flt	flt 3	Sets floating point (scientific notation) format for numeric outputs with the specified number of decimal places (0 to 11).
fxd	fxd 2	Sets fixed point format for numeric outputs with the specified number of digits to the right of the decimal (0 to 11).
grad	grad	Specifies grads will be used for all calculaiions involving angles.

gsb	gsb 15	Goes to the subroutine beginning at the specified line. Upon return, program execution continues at the next line.
	gsb +2 gsb −4	Goes to the subroutine beginning at the specified line relative to the current line.
	gsb "loop"	Goes to the subroutine beginning at the specified label.
gto	gto 23	Goes to the specified line.
	gto +5 gto −1	Goes to the specified line relative to the current line.
	gto "totals"	Goes to the specified label.
if	if A#B	If the statement is true, continues execution on the same line; if the statement is false, the program branches immediately to the next line.
jmp	jmp 2 jmp −4	Jumps the relative number of lines specified, either forward or backward.
	jmp 0	Jumps back to the beginning of the current line.
	jmp X/3	Jumps the relative number of lines indicated by the expression.
list	list	Lists the entire program.
	list 50	Lists the program beginning at the specified line to the end.
	list 12,17	Lists the program between and including the specified lines.
	list k	Lists the special function keys.
lkd	lkd	Disables the live keyboard feature.
lke	lke	Enables the live keyboard feature. The live keyboard feature is enabled when the calculator is turned on.
nor	nor	Clears the master flag so that the program does not stop or trace on flags, but resumes normal program execution.
	nor 5	Clears the master flag and stop and trace flags for the specified line.
	nor 10,20	Clears the master flag and stop and trace flags for the lines between and including the specified lines.

prt	prt "Total=",T	Prints values or text on the calculator printer.
	prt "say""please"""	Quote marks can be printed within text if they are doubled.
rad	rad	Specifies radians will be used for all calculations involving angles.
ret	ret	Returns program execution at the end of a subroutine to the line following the gsb statement which called the subroutine.
sfg	sfg	Sets all flags (0 through 15) to 1.
	sfg A,6	Sets specified flags to 1.
spc	spc 3 spc T	Causes the printer to space the number of blank lines indicated by the expression.
stp	stp	Causes the program to stop.
	stp 5	Sets the master flag and the stop flag for the specified line.
	stp 10,20	Sets the master flag and the stop flags for the lines between and including the specified lines.
trc	trc	Sets the master flag controlling individual line stop and trace flags.
	trc 5	Sets the master flag and the trace flag for the specified line.
	trc 10,20	Sets the master flag and the trace flags for the lines between and including the specified lines.
units	units	Displays the current specification for calculations involving angles: degrees, grads or radians.
wait	wait 2000	Causes the program to wait the specified number of milliseconds before performing the next step, up to about 33 seconds.

FUNCTIONS

abs	abs(Y-X)	Returns the absolute value of the expression.
acs	acs A	Returns the principal value of the arccosine of the expression in the current angular units.
asn	asn T	Returns the principal value of the arcsine of the expression in the current angular units.
atn	atn A	Returns the principal value of the arctangent of the expression in the current angular units.
cos	cos X	Returns the cosine of the expression in the current angular units.
drnd	drnd (Q,2)	Returns the number Q rounded to the specified number of digits. If Q=3.14159, drnd (Q,2)=3.1.
exp	exp(−R)	Returns the naperian e raised to the specified power. exp 1=2.71828.
flg	flg T	Returns 1 if flag is set; 0 if flag is cleared.
frc	frc(A/B)	Returns the fractional part of the expression.
int	int(A/B)	Returns the integer part of the expression.
log	log A	Returns the base 10 (common) logarithm of the expression.
ln	ln A	Returns the base e (natural) logarithm of the expression.
max	max(A,99) max (A[*])	Returns the largest value in the list of expressions and arrays.
min	min(X,Y,0) min(I[*])	Returns the smallest value in the list of expressions and arrays.
prnd	prnd (X,−2)	Returns the number X rounded to the power of ten position specified. If X=3.14159, prnd (X,−2)=3.14.
res	res	Returns the result of the last numeric keyboard operation which was not assigned to a variable.
rnd	rnd T	Returns a pseudo-random number greater than or equal to 0 and less than 1. If T is a negative number, a new seed is generated from the value of T.
sgn	sgn X	Returns: −1 if X is negative, 0 if X is zero, and 1 if X is positive.
sin	sin A	Returns the sine of the expression in the current angular units.
√	√(5X)	Returns the square root of the expression.
tan	tan X	Returns the tangent of the expression in the current angular units.
tn↑	tn↑X	Returns the number ten raised to the specified power.

TAPE CARTRIDGE OPERATIONS

NOTE: If a file number is not specified, file 0 is assumed in all cases.

avd	avd	Disables the automatic tape file verification feature.
ave	ave	Enables the automatic tape file verification feature. Automatic tape file verification is enabled when the calculator is turned on.
ert	ert F	Erases everything on the current track beginning at and including the specified file. It takes about 40 seconds to erase an entire track.
fdf	fdf N	Finds the beginning of the specified file on the current track.
idf	idf A,B,C,D,E idf F idf F,X,X,X,T	Returns identifying information about the file where tape is currently positioned to the specified variables. The first variable receives file number, the second a file type code (see tlist, below), the third the current file size in bytes, the fourth the absolute file size in bytes, and the fifth the current track number. The second through the fifth return variables are optional.
ldb	ldb 3	Loads a binary program into memory from the specified file.
ldf	ldf F	Loads the file specified by F into memory. If the file contains a program, execution continues at the beginning of the program if loaded under program control. If the file contains data, the values are loaded into r-variables beginning with r0 until all data is loaded.
	ldf F,X ldf F,r5	Loads the file specified by F into memory. If the file contains a program, it is renumbered beginning with the line number specified by X, and, if loaded under program control, execution continues at the specified line number. If the file contains data, a value is loaded into the variable specified by X, or, if an r-variable is specified, values are loaded into r-variables beginning with the specified r-variables until all data is loaded.
	ldf F,X,Y ldf F,r5,r9	Loads the file specified by F into memory. If the file contains a program, it is renumbered beginning with the line number specified by X, and execution continues at the line number specified by Y. If the file contains data, values are loaded into the variables X and Y, or, if r-variables are specified, values are loaded into r-variables between and including the specified r-variables.
	ldf F,X,Y,Z ldf A[*],X,I[*]	Loads the values for any number of listed variables, including entire arrays when [*] is indicated.

ldk	ldk K	Loads the special function keys from the specified file into memory.
ldm	ldm F	Loads the entire read/write memory from the specified file into memory.
ldp	ldp 3	Loads the program from the specified file into memory and runs it.
	ldp 7,15	Loads the program and renumbers it beginning at the specified line number and runs it beginning at that line.
	ldp 3,10,5	Loads the program and renumbers it, then runs it beginning at the specified line.
mrk	mrk X,Y	Marks the number of files specified by X, each having the length in bytes specified by Y.
	mrk 5,150,T	Marks the files and returns the last usable file number to the specified variable.
rcf	rcf 3	Records the entire program in memory on the specified file.
	rcf F,50	Records the program lines after and including the specified line number on the file.
	rcf F,10,20	Records the program lines between and including the specified line numbers on the file.
	rcf F,"SE" rcf 3,1,5,"SE"	Secures the recorded program lines so that when the program is later loaded back into memory, it cannot be listed or displayed.
	rcf F,50,"DB"	Includes stop and trace flags for debugging the program on the recorded copy of the program.
	rcf F,A,B,C	Records the values for the specified variables on the file.
	rcf F,X[*]	Records the values for all elements in the specified array on the file.
	rcf F, r2,r50	Records the values for r-variables between and including those specified on the file.
	rcf F,r0	Records the values for all r-variables in memory beginning at the specified r-variable on the file.
rck	rck 7	Records the special function keys on the specified file.
rcm	rcm X	Records the entire read/write memory on the specified file.
rew	rew	Rewinds the tape cartridge. It takes 19 seconds to rewind the entire cartridge.

tlist	tlist	Lists tape cartridge information in the form: Track number File number File type Current file size Absolute size File type codes: 0 - Null or empty file 1 - Binary program 2 - Numeric data file 3 - Data file including String data 4 - Read/Write memory file 5 - Special function key file 6 - Program file
trk	trk X	Sets all tape cartridge activity to the specified track (0 or 1).
vfy	vfy R	Verifies the contents of the file where tape is currently positioned. If the contents of the file is the same as the contents in memory the return variable receives the value 0; if not, the value 1 is returned. This verification of a recording is done automatically when the automatic tape file verification feature is enabled (see ave, above).

STRING ROM

Statements

dim	dim A$[100],B$[100]	Reserves storage space for strings.
	dim A$[N,C]	Space for a string array can be reserved, with the number of strings represented by N, each having the number of characters represented by C.

NOTE: In all of the following statements and functions, a complete string is referred to as A$,B$, etc. When string arrays are used, an extra subscript is needed in each statement or function to indicate which string within the array is being referenced. For example, A$[I] refers to the complete Ith string within the array A$. Substrings therefore also have different appearance when string arrays are used, as follows:

String	String Array	
A$[N]	A$[I,N]	The substring beginning at the Nth character of the string and ending at the end of the string.
A$[N,M]	A$[I,N,M]	The substring between and including the Nth through the Mth characters of the string.

dsp	dsp X,A$,B$[N]	Displays the specified strings or substrings, numbers or constants as listed.
enp	enp"NAME?",A$	Enters and prints values assigned to strings from the keyboard. Prompts (enclosed in quotes) are optional.
ent	ent"COLOR?",C$	Enters values assigned to strings from the keyboard. Prompts (enclosed in quotes) are optional.
if	ifA$#B$	Compares alphabetical (collating) sequence of characters based on standard ASCII codes. If the statement is true, continues execution on the same line; if the statement is false, the program branches immediately to the next line. Any of the relational operators can be used: =, #, >, <, etc.
prt	prt A$,B$,A,B	Prints the specified strings or substrings, numbers or constants as listed.
ldf	ldf F,A$,B$,A,B	Loads the values for the listed variables, including strings and substrings, into memory from the data file indicated by F.
rcf	rcf F,X$,Y$,X,Y	Records the values for the listed variables, including strings and substrings, on the file indicated by F.
→	A$ →B$ "Yes" →C$	Assigns a value to a string or substring. A string or substring may receive its value from another string or substring.

Functions

len	len (A$)	Returns the total character length of the string.
pos	pos (A$,"T") pos (A$,C$)	Returns the position of the specified character or characters in the string. If the character (s) is not found, 0 is returned.
val	val (A$)	Returns the value of the string to be used in computations. Strings themselves cannot be used in computations.
&	A$&B$	Concatenates the strings: If A$ is "AN" and B$ is "DY", A$&B$ returns the string, "ANDY".
num	num(F$[3,3])	Returns the decimal equivalent of the ASCII code for the specified character.
str	str (X)	Converts a numeric value into a string based on the current fxd/flt setting.
char	char (38)	Generates ASCII or non-ASCII characters.
cap	cap (A$)	Converts lower-case alphabetic characters to upper case.

ADVANCED PROGRAMMING ROM

Statements

for	for I=MtoN	Establishes I having the value M. Each time next I is executed, I is incremented by 1, until I reaches a value greater than N.
	for I=MtoNbyX	Establishes I having the value M and increments I by X each time net I is executed, until I has a value greater than N.
next	next I	Causes increment and test of I: for I=M to N by X, I is incremented by X and compared to N. If I is between M and N, the program goes to the statement after the for statement; if I is between M and N, the program goes to the next statement.
xref	xref	Prints a cross-reference listing of program variables and line numbers where used for the program currently in memory.

Subroutine Subprograms

cll	cll 'loop'	Calls the subroutine having the specified label. Upon return, program execution continues at the next statement.
	cll 'loop'(A,B,C)	Calls the subroutine and passes the values of A, B, and C for local use within the subroutine. Within the subroutine the listed parameters are called p-numbers. (A, B, and C will be called p1, p2, and p3, respectively.)

Function Subprograms

'f'	When a function subroutine name appears in single quotes, the program goes to the specified label and continues execution. When ret (return) is encountered, the value is returned to the expression where the function name appears. For example, A+'f'→A.
'SUM' (A,B,C)	Calls the function and passes the values of A, B, and C for local use within the function. Within the function the listed parameters are called p-numbers. The 1st parameter passed is p1, the 2nd is p2, and so on, according to their positions in the parameter list.

ret X — Returns the value of X to the expression containing the function call.

p-numbers

p-numbers may be used within subroutines or functions. Within a function or subroutine p1 is the 1st parameter passed, p2 is the 2nd, and so on. p0 contains the number of passed parameters after the function or subroutine is called. Additional p-numbers beyond those passed are allocated from free memory and are local to the function or subroutine.

Functions

fts	fts (A) →A$[I,I+3]	Changes the full precision number to split precision and stores the packed form of the number in a 4-character field. During packing, only 6-digit accuracy, or numbers within the range ±9.99999E±63, is maintained.
stf	stf (A$[I,I+3]) →X	Unpacks the split precision number produced by the fts function from the 4-character string and returns the value to be used in computations.
fti	fti (A) →A$[I,I+1]	Changes the full precision number to integer precision and stores the packed form of the number in a 2-character field. During packing, only integer accuracy, or numbers wighin the range −32768 and +32767, is maintained.
itf	itf (A$[I,I+1]) →X	Unpacks the integer precision number produced by the fti function from the 2-character string and returns the value to be used in computations.

MATRIX ROM

dim	dim A[50],B[I,25] dim X[1972:1982] dim S[J,K,L]	Reserves space for arrays having the specified dimensions. Multidimensional arrays can be specified, but matrix operations (inv, trn, idn and mat) cannot be performed on arrays having more than 2 dimensions.
ldf	ldf F,A[*]	Loads the values for the entire array A from the file specified by F.
rcf	rcf G,B[*]	Records the entire array B on the file specified by G.
inv	inv A →B inv A →B,D inv A →A	Computes the inverse of matrix A and stores it in matrix B. If return variable D is specified, the determinant of matrix A is returned.
mat	mat A*B →C mat AB →C	Performs matrix multiplication; if matrix A has dimensions m,n and B has dimensions n,p then matrix C has dimension m,p.
trn	trn A→B	Transposes matrix A so that the rows in matrix A become columns in matrix B, and the columns in matrix A become rows in matrix B.
idn	idn A,B,X	The listed matrices become identity matrices. An identity matrix contains all zeros except on the major diagonal, which is all ones. An identity matrix must have the same number of rows as columns.
ara	ara A+B →C ara A−B →C ara A*B →C ara AB →C ara A/B →C ara A →C ara A+B →A	Performs the arithmetic operation element by element on the array. The first element of array A and the first element of array B are added, subtracted, multiplied, or divided, and the result is stored in the first element of array C. Arithmetic operation can be performed on an array in place, and implied multiplication and array copying can also be performed with the ara statement.
smpy	smpy S*A →B smpy SA →B smpy 10A →A	Multiplies each element of the array A by the scalar value S. Implied multiplication is allowed, and the original matrix can be used to store the result.
ina	ina A	Initializes each element of the array A to 0.
	ina A:S,B:10	Initializes array A to the value specified by the variable S.
aprt	aprt A,B	Prints the elements in the specified arrays on the 16-character strip printer.
rdm	rdm A[25],B[25,3]	Redifines the dimensions of the specified arrays, so that the size or shape is altered.

PLOTTER ROM

scl	scl −100,100,0,50 scl A,B,C,D	Establishes the user's scale range for the plotting area: the minimum and maximum values for plotting on the X axis and on the Y axis. The minimum and maximum values on the X axis are represented by A, and B, respectively, and on the Y axis by C and D, respectively.

NOTE: All of the X- and Y-coordinates in the following statements must be within the range of the specified scale statement. Therefore, the scl statement must precede other plotter statements.

axe	axe X,Y	Draws axes through the point X,Y.
	axe X,Y,A axe X,Y,A,B	Tic marks are placed A units apart on the X axis and B units apart on the Y axis, if specified.
pen	pen	Raises the pen.
plt	plt X,Y plt X,Y,P	Moves the pen to specified X,Y point and plots a point. If pen control is used, the pen can be raised or lowered before or after movement to the X,Y point depending on the value of P: Even - Lowers pen Odd - Raises pen Positive - Action before movement to X,Y point. Negative - Action after movement to X,Y point.
ofs	ofs X,Y	Offsets the origin point 0,0 to the point X,Y. For example, two or more functions may be plotted around the point 0,0, each taking only a portion of the paper.
iplt	iplt X,Y iplt X,Y,P	Plots a point the specified number of incremental X and Y units away from the current point. Pen control, specified by P is the same as for the plt statement.
lbl	lbl"SINE X/X" lbl X,"DEGREES"	Prints the specified label on the plotter using the plotter character set. The allowable items in the list are the same as for the standard prt (print) statement.
csiz	csiz H csiz H,A csiz H,A,P csiz H,A,P,R	Establishes the size and shape of characters to be used with the plotter character set. If no csiz statement is used the following default values are used: H – Height of character as a % of total paper height – 1.5% A – Aspect ratio (height/width) – 1 P – Paper ratio (height/width) – 1 R – Angle of rotation – 0 (gives normal left to right lettering).

cplt	cplt X,Y	Plots a point the specified number of character units away from the current point. To center a character over the current pen position, for example, cplt −.3,−.3 is used.
psc	psc 5	Establishes the select code for future outputs to the plotter. If no psc statement is used, the default select code is 5.
	psc 0	If plotter is disconnected or a plotter program is being debugged, all output to plotter is bypassed for testing plotter statements.
ltr	ltr X,Y	Moves the pen to the point X,Y for the purpose of printing a label. Pen is up before movement.
	ltr X,Y,HWD	If HWD is specified, it establishes the relative height, width, and direction of the characters. H and W are integers in the range 1 to 9, and D is in the range 1 to 4, specifying one of the four cardinal directions.
ptyp	ptyp	Establishes plotter typewriter mode. The calculator keyboard becomes a typewriter with the plotter as the output writer. The STOP key terminates plotter typewriter mode.

NOTE: "PLT DOWN" flashes in the display whenever the calculator is not able to send data to the plotter, such as when power to the plotter is turned off.

GENERAL I/O ROM

Statements

fmt	fmt10f6.2	Establishes a list of format specifications to be used for output (with wrt statements) or input (with red statements), and assigns these specs to format number 0.
	fmt 9,3f6.2	Assigns the listed format specifications to the specified format number (0 through 9). The following format specifications can appear in theformat list: f - fixed point numeric e - exponential (scientific) numeric fz - fixed point numeric with leading zeros b - binary code c - character (string) data x - space / - carriage return & line feed (CR/LF) z - suppress CR/LF "text" - quote field
	fmt 4	Cancels previous format specifications assigned to the format number.
wrt	wrt 6	Writes a carriage return and line feed to the specified peripheral select code.
	wrt 6,A wrt 6,"X",X	Writes the specified variables and constants, numbers and strings to the peripheral. The format number 0 specifications are applied to the output; if no format 0 is specified, fmt 4f18 is used for numeric outputs.
	fmt no.↘ wrt 6.3,A wrt 6.9,"X",X	Writes the data to the peripheral, using the format number specified.
red	red S,R red S,A$,A	Reads values into the specified variables from the peripheral specified by select code S. The format number 0 specifications are applied to the input; if no format 0 is specified, numbers are assumed to be separated by commas and terminated by a line feed.
	red 6.3,X,Y	Reads values into the variables from the peripheral using the format specified.
wtb	wtb S,33 wtb S,33,"A",4	Writes the 16-bit binary code or codes to the peripheral specified by select code S. Characters within quotes are output as they appear. CR/LF is not automatically output at the end of the wtb statement.
wtc	wtc 4,64	Writes control bits (not data) to change conditions on the HP 98032A Interface Card or a peripheral.

list #	list #6 list #6,100 list #6,25,50	Lists the entire program or part of the program to the specified select code. Line numbers following the list # parameters are as described under the standard list statement.
	list #6.1	Lists all or part of a program, suppressing automatic CR/LF and checksum.
conv	conv	Cancels previously defined conversion tables.
	conv A,B conv A,B,12,51	Sets up a conversion table for use with red and wrt statements. For both input and output, whenever the character represented by A is encountered the character represented by B is substituted. Up to 10 pairs of characters can be defined.

Functions

rdb	rdb(4)	Returns one 16-bit binary character code from the specified peripheral select code.
rds	rds(6)	Returns the current status conditions (not data) from the specified interface card and peripheral.

EXTENDED I/O ROM

Binary Mode Statements

moct	moct	Establishes octal mode for certain General I/O and Extended I/O ROM operations. In octal mode, all 16-bit oriented parameters are assumed to be expressed as octal numbers.
mdec	mdec	Re-establishes decimal mode (decimal mode is set when calculator is reset). In decimal mode, all 16-bit oriented parameters are assumed to be expressed as decimal numbers.

Bit Functions

ior	ior (A,B)	Returns the 16-bit result of combining A and B, bit by bit, in a logical "inclusive or" operation.
eor	eor (A,B)	Returns the 16-bit result of combining A and B, bit by bit, in a logical "exclusive or" operation.
band	band (A,B)	Returns the 16-bit result of combining A and B, bit by bit, in a logical "and" operation.
cmp	cmp A	Returns the 16-bit complement of A, taken bit by bit, in a logical "not" operation.
rot	rot (A,P)	Returns the 16-bit result of binary right rotation of the bits in A, the number of positions indicated by P. If P is negative, the bits are rotated P places to the left.
shf	shf (A,P)	Returns the 16-bit result of binary shift to the right of the bits in A, the number of positions indicated by P. If P is negative, the bits are shifted P places to the left.
add	add (A,B)	Returns the result of addition of A and B. If octal mode (moct) is in effect, octal numbers are added, giving octal results. In decimal mode, results are the same as for A+B.
bit	bit (N,A)	Returns the value of the Nth bit in A(0 or 1).
	bit ("101XX",A)	Returns 1 if the mask matches the bit pattern in A, or 0 if the mask does not match. X or other character in the mask indicates bit which is not checked.
dto	dto A	Returns the octal equivalent of the decimal value specified by A.
otd	otd A	Returns the decimal equivalent of the octal value specified by A.

HP-IB Control Statements

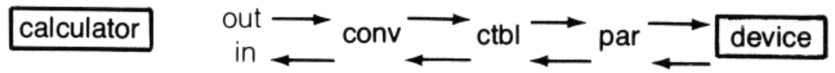
dev	dev "punch",703 dev "scan",715	Establishes a user name for a peripheral device. Once established, device names can be used in place of select codes.
cmd	cmd 7,"?U$","L10" cmd "scan","L10" cmd 7,"?U$","home" cmd "scan","home"	Sends the first set of characters to specify addresses and the second set of characters to instruct the device. A device (dev) name can be used to set up the bus, and an equate (equ) name can be used to instruct the device.
equ	equ "home","L10"	Equates the sequence of ASCII characters or codes with a user name, for use with the cmd statement.
trg	trg 7 trg 711	Sends a trigger message (GET) to the specified device.
clr	clr 7	Sends Clear message (DCL) to all devices.
	clr 711	Sends Selective Clear (SDC) to device.
cli	cli 7	Sends Abort message (IFC) to all devices.
rem	rem 7	Sends Remote message (REN) to all devices.
	rem 711	Sends Remote message to selected device.
lcl	lcl 7	Sends Local message (GTL) to all devices.
	lcl 711	Sends Clear Lockout/Local message ($\overline{\text{REN}}$) to selected device.
llo	llo 7	Sends Local Lockout message (LLO) to all devices.
polc	polc S,B	Sets parallel-poll bits on device S.
polu	polu S	Clears parallel-poll bits on device S.
pol	pol 7	Conducts a parallel poll on bus.
pct	pct 721	Pass bus control to device 21 on bus.
rqs	rqs 7,5	Request Service from controller and send serial byte 5 for response to a serial poll.
rds	rds (711)	Conduct serial poll from device 11 on bus.

Conversion Statements

ctbl	ctbl A$	Establishes a conversion table using a string for all General I/O and Extended I/O input and output operations. The value of each string character represents ASCII, while the position of each character represents the value of the foreign code. ctbl cancels the conversion table.
par	par X	Establishes the parity to be used when data is output and to check data on input. The possible values of X are: 0, parity not checked; 1, parity always one; 2, even parity; and 3, odd parity.

Conversion Protocol

When more than one conversion statement is in effect, they are referenced in this order:



Interrupt Statements

oni	oni S,"shutoff"	Establishes the service routine where program execution will branch to interrupt from the peripheral specified by select code S.
eir	eir S	Enables the calculator to accept an interrupt from the peripheral specified by select code S.
	eir S,M	Enables interrupt according to byte M. eir S,0 disables interrupt.
iret	iret	Signals the end of a service routine. During the interrupt service routine, the interrupt for the peripheral being serviced is automatically disabled to prevent cascading of interrupts.

Error Recovery

on err	on err "restart"	Establishes a routine to be followed in case a calculator error condition occurs. The following three functions are assigned values when the specified error recovery routine is reached.
	rom	The ROM in which the error occurred. 0=mainframe error. Other ASCII-decimal numbers indicate letter of plug-in ROM.
	ern	The error number.
	erl	The line in which the error occurred.

Buffered I/O Statements

buf	buf "prt",A,T buf "dvm"A$,T	Establishes a read/write buffer, in either a special memory area or a string, for buffered interrupt, DMA and Fast Read/Write operations. The buffer name can be used in place of select codes for I/O operations. The type of buffer is established by the code specified in T. The buffer types are:

	Word Oriented	Byte Oriented
Interrupt	0	1
Fast Read/Write	2	3
DMA	4	—

tfr	tfr "prt",D	Transfers data from buffer "prt" to device D.
tfr	tfr D,"dvm"	Transfers data from device D to buffer "dvm".
tfr	tfr D, "dvm" ,X	Transfers X number of words or bytes from device X to buffer "dvm".
tfr	tfr D, "dvm" ,X,A	Transfers data until either X number of words (or bytes) are input or character A is input.

Time Statement

time	time T	Specifies a maximum limit of time (in milliseconds) to wait for a device to become ready before exiting the I/O operating and displaying error E4.

Interface Control Statement

wti	wti R,A	Writes A to the control register R on the interface Card.

Interface Control Functions

rdi	rdi (R)	Returns the value from the control register specified by R on the interface card.
iof	iof (S)	Examines the flag line on the interface card. Returns 1 if peripheral is ready or 0 if peripheral is busy.
ios	ios (S)	Examines the status line on the interface card. Returns 1 if interface card is functional or 0 if interface card is in error condition.

FLEXIBLE DISK ROM

The HP 9885 Disk Drive is a mass storage device that uses a flexible disk as the storage medium. Information stored on flexible disks can be accessed much faster than information stored using tape cartridges. In addition, disks have more than twice the storage capacity of tape cartridges. The flexible disk system can accommodate up to eight 9885M Drives. (An HP 9878A I/O Expander is required when more than three I/O devices are connected to a calculator, including 9885 Drives.) Each 9885M can accommodate up to three additional 9885S Drives.

Disk Capacity and Speed

468,480 bytes of storage space per disk
1830 records (256 bytes/record) per disk
352 files maximum per disk
360 revolutions per minute
267ms average access time
23,000 bytes/second transfer rate for numerics

Disk ROM Statements

All disk statements are programmable. In addition, all disk statements are executable from the keyboard and in the live keyboard mode, except for `get`, `chain` and `getk` which are not allowed in live keyboard mode.

drive	drive 0,8 drive X,Y	Specifies the drive (0 thru 3) to be used and optionally, the select code (8 thru 15), indicating the 9885M drive being addressed. Default values are 0,8.
cat	cat cat 0	Outputs abbreviated information to the calculator printer about all user files on the disk, including – File name File type File size – in bytes for program, binary, memory or key files and in records for data files. Number of available records and bootstraps version. File type codes – P - Program file K - Key file D - Data file M - Memory file B - Binary program file O - Other

	cat 16	Outputs complete information to the calculator printer about all user files on the disk, including information listed above, plus – File location (Number of records used and location by Track and Record number.)
	cat 9 cat X	Outputs complete information about all user files to the selected device when a select code (2 thru 15) follows cat.
	cat "buffer"	Outputs complete information about all user files to the calculator buffer when a buffer name (string) follows cat.
save	save "Name" save N$,X save N$,X,Y	Stores the entire program, or the lines between and including the specified line numbers, in the file named.
get	get "Name" get N$	Loads the program specified from the disk into the calculator memory. (Variable values are lost.)
	get "Name",5 get N$,X	Loads the program specified from the disk and renumbers it starting with the line number specified by X. (Variable values are lost.)
	get "Name",5,99 get N$,X,Y	Loads the program specified from the disk, renumbers it starting with the first line number specified and automatically executes it from the second line number specified. (Variable values are lost.)
chain	chain "Name",5,99 chain N$,X,Y	Loads the program specified from the disk into the calculator memory and retains the values of all variables. Same line number rules apply as for the `get` statement.
resave	resave "Name" resave N$,X,Y	Stores a new program, or the lines indicated by the line numbers, on the disk using a previous file name. Same line number rules apply as for the `save` statement.
savek	savek "keys" savek K$	Stores all present special function key definitions in the named file on the disk.
getk	getk "keys" getk K$	Loads all special function key definitions from the specified file of the disk to the calculator special function keys.
open	open "Data",10 open D$,10	Creates a data file on the disk with the indicated number of records and assigns it the name specified. End of file (EOF) marks are written in the beginning of each record.
kill	kill "Data" kill D$	Erases the program, data, memory, binary or key file specified from the disk and makes the file space available.

files	files Data:1, Names:1	Assigns file numbers (1 thru 10) to the files named and indicates optional drive numbers for each file.
	files *	An asterisk may be used for the file name, or names, if an asgn statement follows, to reserve a space for a file or to enable use of a string variable for a file name, or names.
asgn	asgn "Data", 2,0,X asgn D$,F,D, X	Assigns a file number (1 thru 10) to a single file (F) and indicates optional drive number (D) for the file specified. A return variable (X) can be used for further file information. Variable Value 0 - file is available and assigned 1 - file doesn't exist 2 - program file 3 - key file 4 - file type not defined 5 - memory file 6 - binary program file 7 - file type not defined 8 - file number out of range
sprt	sprt 1,A,B$,C[*] sprt F,A,B$,C[*]	Prints specified data items in the file number (F) indicated after the last item printed or read. An end of record (EOR) mark is printed after all data stored.
	sprt 1,A,B$,"end" sprt F,A,B$,"end"	If "end" is the last parameter, an EOF mark is printed after all data stored.
	sprt 1,A,"ens" sprt F,A,"ens"	If "ens" is the last parameter, no EOR or EOF mark is printed; for this reason the "ens" parameter should be used with care.
sread	sread 1,A,B$,C[*] sread F,A,B$,C[*]	Reads data from the specified file (F) starting after the last item printed or read.
rprt	rprt 1,1,A,B$,C[*] rprt F,R,A,B$,C[*]	Prints specified data items in the file number (F) indicated starting at the beginning of the record number (R) indicated and prints an EOR mark at the end of all data.
	rprt 1,1,A,B$,"end" rprt F,R,A,B$,"end"	Prints specified data items in the file number (F) indicated starting at the beginning of the record number (R) indicated and prints an EOF mark at the end of all data.
	rprt 1,1,A,"ens" rprt F,R,A,"ens"	Prints specified data items in the file number and record number indicated without printing an EOR or EOF mark; for this reason it should be used with care.
	rprt 1,1,"end" rprt F,R,"end"	Erases the specified record by placing an EOF mark in the beginning of it.

rread	rread 1,1,A,B$,C[*] rread F,R,A,B$,C[*]	Reads data from the specified file (F) starting at the beginning of the specified record (R).
	rread F,R	Repositions the file pointer to the beginning of the specified record in the file indicated.
type	type X	Identifies the type of the next item in a specified file.
	type-X	Used to detect an EOR mark. Type codes 0 - Type unknown, undefined 1 - full precision number 2 - string that doesn't overlap record boundaries 2.1 - first part of a string that overlaps record boundaries 2.2 - intermediate part of a string that overlaps record boundaries 2.3 - end part of a string that overlaps record boundaries 3 - end of file (EOF) mark 4 - end of record (EOR) mark
renm	renm "Old", "New" renm O$,N$	Changes the name of a file from the original name (O$) to the new name (N$) specified.
on end	on end 3,55 on end 3, "Label"	Sets up a branching condition which changes the program flow to a specified new location (by line number or label) when an EOF mark is encountered during an sprt or sread or an EOR or EOF mark is encountered during an rprt or rread.
savem	savem "Memory" savem M$	Stores the calculator's entire read/write memory into the file specified.
getm	getm "Memory" getm M$	Loads the calculator's entire read/write memory from the specified file and returns the calculator to its state before savem was executed.
copy	copy 0,8, "to",1,8 copy X,S,"to",Y,S	Duplicates the entire contents of a specified source disk (X) to a specified destination disk (Y) which has as many or more usable tracks as the source disk, using optional select codes (S), if necessary.
	copy "Old",0,8, "New",0,8 copy O$,D,S,N$, D,S	Duplicates the contents of a specified source file (O$) into the specified destination file (N$) indicating optional drive numbers (D) and select codes (S), if necessary.
	copy 1,1,8,1,10 copy O,R,N,T,X	Duplicates a specified source file (O), beginning at the indicated record number (R), into the specified destination file (N), beginning at the indicated record number (T), for the number of records specified (X).
dump	dump X dump-X (where X= 1 or 10)	Stores the entire disk onto up to three tape cartridges. Optional parameters, if negative, suppress automatic tape marking; 1 or 10 indicates the number of disk records to be stored per tape file.

	dump "File",8,X dump F$,F,−X (where X=1 or 10)	Stores named data file from the disk into the specified tape file. Optional parameter same as for dump disk.
load	load	Loads entire disk from the tape files on the tape cartridges used to dump the disk starting with track 0, file 0 of the first tape.
	load "File",8 load F$,F	Loads data from the specified tape file (F) to the disk file named (F$).
repk	repk	Rearranges user files on the disk for more efficient use of available space and for faster statement execution.
von	von	Enables the verify mode which does a read after write under stricter specifications and compares what was read with what's in memory.
voff	voff	Disables the verify mode, as when the system is turned on.
getb	getb "Name" getb N$	Loads a binary program from the disk into the binary program area of the calculator memory.

The following statements are available using the binary programs on the Disk System Cartridge. See the Disk Programming Manual for further information.

init	Loads initialization routine and bootstraps.
boot	Loads bootstraps only.
vfyb	Verifys bootstraps.
killall	Kills all user files only.
dtrk	Dumps a bad track during Error Recovery Routine.
tinit	Reinitializes a bad track during Error Recovery Routine.
ltrk	Returns corrected data to a reinitialized track during Error Recovery Routine.
dirc	Copies the spare directory into the main directory.

9872A Plotter ROM

The HP 9872A Plotter can be used to provide hard copy graphic solutions to problems solved by the HP 9825A. In addition to plotting, the 9872A can be used to draw axes with or without tic marks and labels, to label alphanumerics and symbols, and to provide digitized data to the calculator. It is connected to the calculator via the HP 98034A HP-IB Interface.

The 9872A Plotter ROM uses 104 bytes of user Read/Write Memory when installed.

Plotter Default Conditions

Plotter select code	705 (not changed with pclr or "DF")
Scale	Centimeter unit of measure from P1 (not changed with pclr or "DF")
Line type	Solid line
Line pattern length	4% of the distance from P1 to P2
Limit	Total platen area
Character size	1.5, 2, 1, 0
Automatic pen pickup	On
Pen velocity	36 cm/sec
Adaptive pen velocity	Off
Symbol mode	Off
Tic length	.5% of /P1-P2/ length for each half
Standard character set	Set 0 (Set 1 for pclr)
Alternate character set	Set 0
Character slant	0°
Mask value	223,0,0

Plotter ROM Statements

scl	scl X_{P2}, X_{P1}, Y_{P1}, Y_{P2} scl −100,100,0,50	Locates the origin (point 0,0) and specifies the scale units to be used for plotting by assigning X and Y coordinate values to the scaling points P1 and P2.
psc	psc 806 psc 1515	Specifies the interface select code (one or two digits; range 2 through 15), and plotter address (two digits; range 0 through 30). psc 705 is set at power on, reset and erase a.
	psc 0	All plotter ROM statements in a program are bypassed for use in testing program statements.
pclr	pclr	Sets all parameters that have been sent to the plotter to their default values except scale units, select code, P1, P2, pen location, and pen selection.

xax

```
xax 5
xax 5,1
xax 5,1,-10
xax 5,1,-10,10
xax 5,1,-10,10,2
```

Draws a horizontal axis which crosses the Y axis at the specified coordinate.

If tic marks are to be drawn, the parameter specifies the spacing in scale statement units.

The start and end points of the axis can also be specified. Labels, and the number of tic marks between them, are specified by the last parameter. If negative, neither the axis or tic marks are drawn. If 0, no labels are drawn.

yax

```
yax 0
yax 5,1
yax 5,1,-10
yax 5,1,-10,10
yax 5,1,-10,10,2
```

Draws a vertical axis which crosses the X axis at the specified coordinate. Tic marks, ends of the axis and labels are specified as in the xax statement.

pen

```
pen
```

Raises the pen without moving it to a new location.

pen#

```
pen# 2
pen#
```

Selects the pen to be used. The range of the number is 0 through 4. 0 or no number returns the current pen to a storage location.

plt

```
plt X,Y,P
plt 2,2
plt 2,2,P
```

Moves the pen to the specified X,Y coordinate position. Pen control is determined by the value of P:

Even: Lowers pen
Odd: Raises pen
Positive: Action before movement
Negative: Action after movement
0: No change

ofs

```
ofs X,Y
ofs 4,4
```

Moves the origin (point 0,0) by the number of scale units specified by the incremental values of X and Y.

iplt

```
iplt -3,5
iplt -2,2,P
iplt X,Y,P
```

Moves the pen to a point that is the specified number of X and Y units away from the current point. Pen control, specified by P, is the same as for the plt statement.

line

```
line
line 4
line 4,10
```

Specifies the type of line that will be uses with plt, iplt, xax and yax statements. A solid line is specified by no parameter. The range of the pattern number is 0 through 6. The range of the pattern length (one segment; percentage of diagonal distance between P1 and P2) is ±127.999949999; 4% is default.

lim

```
lim
lim -4,4,-5,2
lim XLL, XUR, YLL, YUR
```

Restricts programmed pen movement to a specific rectangular area on the platen. This area is called the "window". The parameters specify the X and Y coordinates of the lower left and upper right corners of the window. If the parameters are omitted, the window is automatically set to the mechanical limits of the plotter.

lbl

```
lbl "9872"
lbl X,X+1
lbl "X=", X
lbl A$
```

Allows characters to be lettered on the plotter. It is used like the prt statement to letter text, expressions or string variables (a string variables ROM is required to letter strings).

csiz

```
csiz
csiz 5
csiz 5,2
csiz 5,2,2/3
csiz 5,2,2/3,45
```

Specifies the size, shape and direction of characters which are to be lettered. The parameters (in order) specify height (% of scale height), aspect ratio (height/width), paper ratio (height/width) and angle of rotation (in current angular units).

The default values, set when any or all parameters are omitted, are:

Height: 1.5%
Aspect ratio: 2
Paper ratio: 1
Angle of rotation: 0

cplt

```
cplt
cplt 20,-5
```

Moves the pen the specified number of character space fields: widths and heights. If no parameters are specified, the pen is moved one character height down and to the left margin (defined by the last plt, iplt or front-panel operation).

ptyp

```
ptyp
```

Sets a manual lettering mode. All characters typed on the keyboard are lettered on the plotter. To end the ptyp mode, press the stop key. The following keys perform these functions in ptyp mode:

- → space
- ← backspace
- ↓ linefeed
- ↑ inverse linefeed
- STORE carriage return

dig	dig A,B dig A,B,P	Enables the digitizer mode; this lights the ENTER light on the plotter. Pressing ENTER then assigns the X and Y coordinate values of the pen position to the specified variables, respectively. The optional third variable returned specifies pen status: 0 = up 1 = down To cancel a digitize statement without entering values, press the STOP key.

Systems Programming ROM

The 98224A Systems Programming ROM extends the 9825A language to include capabilities for remote keyboard operation, program self-modification, run-time read/write storage allocation, and intelligent terminal emulation. It uses 160 bytes of user read/write memory and can't be used concurrently with the 98211 Matrix ROM.

Systems Programming ROM Statements

on key	on key "kbd" on key "kbd", 7	Enables keyboard interrupt capability and specifies a service routine name. A flag can optionally be specified to be set if the key buffer overflows.
	on key	Without parameters, the on key statement disables the service routine, clears the key buffer and returns the calculator to normal keyboard operation.
kret	kret	Returns program execution to the main program only after emptying the key buffer.
eol	eol 13 eol 13,10,32,32	Specifies one to seven characters to be used as the end-of-line sequence, replacing all General I/O end-of-line delimiters.
	eol13,10,32,32, −1000	Specifies the delay (in milliseconds) between output of the last character of the eol sequence and the start of the next line of output.
wsc	wsc 6,39	Specifies the R4D USART control word at the specified select code. The range of the select code is 2 through 15 and must specify a 98036 Serial Interface.

wsm	wsm 6,123	Specifies a mode word to be written to the R4C register of the 98036 Serial Interface at the specified select code.
	wsm 6,123,39	A control word can also be added to the mode word syntax. The default value is 5 if not specified.
rkbd	rkbd 11	Enables a remote keyboard at the specified select code to control the calculator, without disabling the 9825A keyboard.
	rkbd 11,1	The code type, 1 or 0, specifies 9825A keycode-type keyboard or ASCII-type keyboard, respectively. Zero (ASCII) is the default.
store	store A$ store "prt A" store A$,5 store "prt A",nal	Stores program lines from an executing program. The line number determines which program line the line will be stored at. The String ROM must be installed to use a string variable.

Functions

key	key	Returns the earliest entered unprocessed keycode in the key buffer. Zero is returned when all keycodes have been processed.
asc	asc 38 asc key	Returns the ASCII equivalent of a 9825A keycode.
bred	bred ("Buff")	Returns the contents of the specified buffer. The buffer specified must be an active, interrupt type, byte input buffer (Extended I/O type 1).
rss	rss 6	Returns the contents of the 98036A status register (register R4E).
nal	nal	Returns the value of the last program line number plus one. When used with the store statement, it overrides the line number profix of the string to be stored.

avm	avm	Returns the number of unused bytes remaining in the 9825A's read/write memory.
cln	cln	Returns the value of the current line number at the point of execution.
Miscellaneous		
%	% text	The free text prefix is used to store all text following it into memory without any syntax checking. Free text protection is terminated by a semicolon or new program line.

Mainframe Error Messages

An error in a program sets the program line counter to line 0. Pressing the continue key will continue the program from line 0. Execute the continue command with a line number to continue at any desired line (such as: `cont 50`).

error	Description
00	System error.
01	Unexpected peripheral interrupt.
02*	Unterminated text.
03*	Mnemonic is unknown.
04	System is secured.
05	Operation not allowed; line cannot be stored or executed with line number.
06*	Syntax error in number.
07*	Syntax error in input line.
08	Internal representation of the line is too long (gives cursor sometimes).
09	gto, gsb, or end statement not allowed in present context[1].
10*	gto or gsb statement requires an integer.
11	Integer out of range or integer required. Must be between −32768 and +32767.
12*	Line cannot be stored; can only be executed.
13	Enter (ent) statement not allowed in present context.
14	Program structure destroyed.
15	Printer out of paper or printer failure.

[1] See also Advanced Programming ROM Error Messages.

* These errors give a cursor when the RECALL key is pressed, indicating the location of the error in the line.

16 String Variables ROM not present for the string comparison. Argument in relational comparison not allowed.

17 Parameter out of range.

18 Incorrect parameter.

19 Bad line number.

20 Missing ROM or binary program. The second number indicates the missing ROM. In the program mode, the line number is given instead of the ROM number.

Number in Display	ROM
1	Binary Program
4	Systems Programming
6	Strings
8	Extended I/O
9	Advanced Programming
10	Matrix
11	Plotter (9862A or 9872A)
12	General I/O
15	9885 Disk

21 Line is too long to store.

22 Improper dimension specification.

23 Simple variable already allocated.

24 Array already dimensioned.

25 Dimensions of array disagree with number of subscripts.

26 Subscript of array element out of bounds[1].

27 Undefined array.

28 Ret statement has no matching gsb statement.

29 Cannot execute line because a ROM or binary program is missing.

[1] See also Advanced Programming ROM Error Messages.

30 Special function key not defined.

31 Non-existent program line.

32 Improper data type[1].

33 Data types do not match in an assignment statement.

34 Display overflow due to pressing a special function key.

35 Improper flag reference (no such flag).

36 Attempt to delete destination of a gto or gsb statement.

37 Display buffer overflow caused by display (dsp) statement.

38 Insufficient memory for subroutine return pointer[1].

39 Insufficient memory for variable allocation or binary program.

40 Insufficient memory for operation[1].

41 No cartridge in tape transport.

42 Tape cartridge is write protected. (Slide record tab to other position for recording.)

43 Unexpected Beginning-Of-Tape (BOT) or End-Of-Tape (EOT) marker encountered; or a tape transport failure.

44 Verify has failed.

45 Attempted execution of idf statement without parameters or mrk statement when tape position is unknown.

46 Read error of file body. (See Appendix F.)

47 Read error of file head. (See Appendix F.)

48 End-Of-Tape (EOT) encountered before specified number of files were marked.

49 File too small.

50 Ldf statement for a program file must be last statement in the line.

[1] See also Advanced Programming ROM Error Messages.

51	A ROM is present but was not when the memory was recorded. Remove the ROM indicated by the number to the right of the error number in the display, and re-execute the ldm statement. In the program mode, the line number is given instead of the ROM number. See error 20 for a list of ROM numbers.
52	The ROM indicated by the number to the right of the error number was present when the memory was recorded but is now missing. Insert the indicated ROM and re-execute the ldm statement. See error 20 for a list of ROM numbers.
53	Negative parameter in cartridge statement.
54	Binary program to be loaded is larger than present binary program and variables have been allocated.
55	Illegal or missing parameter in one of the cartridge statements.
56	Data list is not contiguous in memory for one of the cartridge statements.
57	Improper file type.
58	Invalid parameter in rcf statement: "SE" or "DB" expected.
59	Attempt to record a program or special function keys which do not exist.
60	Attempt to load an empty file or the null file (type = 0).
61	The line referenced in an ldf or ldp statement does not exist. If the line containing the ldf or ldp statement has been overlaid by the load operation, the line number in the display may be incorrect.
62	Specified memory space is smaller than cartridge file size.
63	Cartridge load operation would overlay subroutine return address in program; load not executed.
64	Attempt to execute ldk, ldf (program file), or ldp during live keyboard statement.
65	File not found, or file specified in the previous fdf statement does not exist.

Default values associated with errors 66 through 77 when you set flag 14 are explained in Chapter 5 of the operating and programming manual.

66	Division by zero. A mod B, with B equal to zero.
67	Square root of negative number.
68	Tan (n*π/2 radians); Tan (n*90 degrees) Tan (n*100 grads); where n is an odd integer.
69	Ln or log of a negative number.
70	Ln or log of zero.
71	Asn or acs of number less than −1 or greater than +1.
72	Negative base to a non-integer power.
73	Zero to the zero power (0↑0).
74	Storage range overflow.
75	Storage range underflow.
76	Calculation range overflow.
77	Calculation range underflow.

Advanced Programming ROM Error Messages

error	Description
A0	Relational operator in for statement not allowed. No closing apostrophe.
A1	A for statement has no matching next statement.
A2	A next statement encountered without a previous for statement.
A3	Non-numeric parameter passed as a p-number.
A4	No return parameter for a function call.
A5	No functions or subroutines running. Improper p-number.
A6	Attempt to allocate local p-numbers from the keyboard.
A7	Wrong number of parameters in fts, stf, fti, or itf function. stf or itf parameter must be a string (not a numeric). stf or ift parameter contains too few characters.
A8	Overflow or underflow in fts function or overflow in fti function.
A9	String Variables ROM missing for stf or itf functions.

These mainframe errors have an additional meaning with the AP ROM installed.

09	Attempt to execute a next statement from keyboard while for/next loop using same variables is executed in program or from program while for/next loop using same variable is executed from keyboard. Attempt to call function or subroutine from keyboard.
26	P-number reference is negative.
32	Non-numeric value in for statement or in fts or fti function.
38	Memory overflow during function or subroutine call.
40	Memory overflow while using for statement or while allocating local p-numbers.

Systems Programming ROM Error Messages

error	Description
C0	Missing General or Extended I/O ROM.
C1	Incorrect number of parameters.
C2	Improper parameter specified.
C3	Wrong parameter type.
C4	Illegal buffer type for bred statement.
C5	Key buffer overflow.
C6	Too large or wrong sign of parameter.
C7	Improper execution of store statement.
C8	Illegal use of kret.
C9	Missing 98036A Interface card.

Extended I/O ROM Error Messages

error	Description
E0	General I/O ROM missing. HP-IB error under interrupt.
E1	Wrong number of parameters
E2	Improper buffer device or equate table usage. Multiple-listeners error. Buffer busy.
E3	Wrong parameter type.
E4	Timeout error.
E5	Buffer underflow or overflow.
E6	Parameter value out of range.

E7	Parity failure.
E8	Improper use of iret statement. Attempt to DMA with an HP-IB. Buffer or select code is busy.
E9	Illegal HP-IB operation.

General I/O ROM Error Messages

error	Description
G1	Incorrect format numbers.
G2	Referenced format statement has an error.
G3	Incorrect I/O parameters.
G4	Incorrect select code.
G5	Incorrect read parameter.
G6	Improper conv statement parameters.
G7	Unacceptable input data.
G8	Peripheral device down.
G9	Interface hardware problem.

Matrix ROM Error Messages

error	Description
M1*	Syntax error.
M2	Improper dimensions. Array dimensions incompatible with each other or imcompatible with the stated operation.
M3	Improper redimension specification: New number of dimensions must equal original number; new size cannot exceed original size.
M4*	Operation not allowed. An array which appears to the left of ' cannot also appear on the right.
M5	Matrix cannot be inverted. Computed determinant = 0.

9862A Plotter ROM Error Messages

error	Description
P1	Wrong state. Statements executed out of order. (See Appendix in ROM manual.)
P2	Wrong number of parameters.
P3	Wrong type of parameters. Parameters for a label statement must be expressions, text, or string variables.
P4	Scale out of range. Maximum value is less than or equal to the minumum value.
P5	Integer out of range. Pen control parameter is out of the range −32768 to +32767 or the select code is not 0 or in the range of 2 through 15.
P6	Character size out of range. Width or height in letter statement is zero or there is an integer overflow in csize calculations or results.
P7	Not used.
P8	Axes origin off-scale. X, Y specified for axis statement doesn't fall on plotter surface.

An explanation of the error message PLT DOWN is in the manual.

*These errors give a cursor when the RECALL key is pressed, indicating the location of the error in the line.

9872A Plotter ROM Error Messages

error	Description
P1	Attempt to store into constant. Occurs when one or more parameters in a dim instruction are constants rather than variables.
P2	Wrong number of parameters. Occurs on instructions with numeric-only parameter lists (scl, ofs, plt, iplt, cplt, xax, yax, lim, dig, csiz, line, pen#, and psc). In certain unusual cases where a parameter list contains user-level function calls, an instruction having an incorrect number of parameters may be executed.
P3	Wrong type of parameter or illegal parameter value.
P4	No HP-IB device number specified. Occurs on a psc instruction when the parameter is between 0 and 14 inclusive and an HP-IB card is at the corresponding select code.
P5	Pen control value not in −32768 thru 32767 range. Occurs on plt and iplt. May also occur if hardware transmission error occurs between plotter and calculator.
P6	No HP-IB card at specified select code. Occurs on psc instruction when the interface card set to the specified select code is not an HP-IB card.
P7	axe, ltr instructions executed. Occurs on axe and ltr instructions because the ROM recognizes these instructions but cannot execute them. This error flags all axe and ltr instructions for the purpose of converting 9825/9872 programs.
P8	Calculator STOP key cancelled operation. Occurs on any instruction when the plotter fails to respond for 3 seconds after the STOP key has been pressed. This error is most likely to occur when the pen is traveling slowly.

P0	Transmission error. The calculator has received an illegal ASCII input from the plotter.
P1	Instruction not recognized. The plotter has received an illegal character sequence.
P2	Wrong number of parameters. Too many or too few parameters have been sent with an instruction.
P3	Bad parameter. The parameters sent to the plotter with an instruction are out of range for that instruction.
P4	Illegal character. The character specified as a parameter is not in the allowable set for that instruction.
P5	Unknown character set. A character set out of the range 0 thru 4 has been designated as either the standard or alternate character set.
P6	Position overflow. An attempt to draw a character or perform a cplot that is located outside of the plotters numeric limit of −32768 to +32767.

Error messages generated by write (wrt) and read (red) statements will typically be displayed as an error in the next executed plotter ROM statement. This can be avoided by using an output error command (wrt select code, "OE") followed by a read statement (red select code, variable) to check for errors after read or write statements that address the plotter.

String Variables ROM Error Messages

error	Description
S0	Invalid set of strings in data list of load fille (ldf) statement.
S1	Improper argument for string function or string variable.
S2	More parameters than expected for string function or string variable.
S3	Accessing or assigning to non-contiguous string. num function of null string.
S4	Trying to find the value of non-numeric string or null string. Exponent too large. Exponent format invalid (e.g., 1e+ +5).

S5	Invalid destination type for string assignment.
S6	Parameter is zero, negative, exceeded dimensioned size. Invalid sequence of parameters for string variable.
S7	String not yet allocated.
S8	String previously allocated.
S9	Maximum string length exceeded; additional string length must be specified in dim statement.

Disk ROM Error Messages

Hardware Errors

error	Description
d0	Firmware/driver out of synchronization. More than six defective tracks in a row. (Press RESET)
d1	All drives in system not powered.
d2	Door opened while disk is being accessed.
d3	Disk not in drive or no such drive number.
d4	Write not allowed to protect disk.
d5	Record header error. (Use Error Recovery Routine.)
d6	Track not found. (Use Error Recovery Routine.)
d7	Data Checkword error. (Use Error Recovery Routine.)
d8	Hardware failure. (Press RESET)
d9	Verify error due to drive problem. Marginal data. (Reprint data)

Software Errors

error	Description
D0	Improper argument.
D1	Argument out of range.
D2	Improper file size (negative, 0 or >32767).
D3	Invalid file name.
D4	File not found.
D5	Duplicate file name.
D6	Wrong file type.
D7	Directory overflow.
D8	Insufficient storage space on disk.
D9	Verify error due to cable, calculator or drive problem. Bad data (Reprint data.)
F0	File overflow when read or print executed.
F1	Bootstraps not found. (Reload bootstraps)
F2	String read but wrong data type encountered.
F3	Attempt to read data item but type doesn't match.
F4	Availability table overflow. (Repack)
F5	Attempt on end branch from other than running program.
F6	Unassigned data file pointer.
F7	Disk is down so line cannot be reconstructed.
F8	Disk is down and (STOP) pressed.
F9	System error. (Save files individually and reinitialize.)

These errors may result during the binary Initialization and Error Recovery Routines.

B0	Wrong syntax, argument out of range or variable not properly dimensioned.
B1	More than six defective tracks on the disk.
B2	Verify error. Boots on the disk not identical to boots on the cartridge.
B3	dtrk or tinit not allowed because error information lost or error not d5, d6, d7 or d9.
B4	Attempt to access record for error correction which isn't part of data file.
B5	Improper string length (inconsistent with length given in header).
B6	Not enough space in calculator buffer for data item or item can't be placed in this part of buffer.
B7	Missing Disk or String ROM.
B8	Track still bad after tinit.

These mainframe errors take on additional meaning when the Disk ROM is installed.

03	Mnemonic not found because disk may be down.
29	Line can't be executed because ROM (usually String) is missing.
31	Line not found.
50	Get or chain should be last statement in a line.
51	ROM now installed which wasn't when savem was executed.

52	ROM now missing which wasn't when savem was executed.
63	Disk load operation would overlay gsb return address so load not executed.
64	Get, chain or getk not allowed from live keyboard mode or during an ent statement.
SPARE DIR.	is printed when the spare directory in the backup track automatically replaces the main directory.
DISK IS DOWN	is displayed when running a program that uses a drive number of a drive that is not connected to the system, not powered or whose door is opened.