HP-67 Quick Reference

Memory & Display

Memory	26 storage registers, 224 program steps. Memory not preserved thru power-off
FIX	Select fix point format
SCI	Select scientific (exponential) format
ENG	Select engineering (exponential) format where the exponent always is a multiple of 3
DSP n	Select number of decimal digits, n=0...9
DSP (i)	Select number of decimal digits according to the absolute value of the integer part of the value in the index register I
RND	Round X to the current number of display digits
CLx	Clear X register

Storage Registers & Indirect Operations

General	Storage registers are: R0-R9, A-E, & I (primary registers) There's also a secondary register set R10-R19 but its registers cannot be accessed directly thru STO & RCL! Rather, indirect addressing or P↔S must be used
CL REG	Clears registers 0-9, A-E & I. Does not clear secondary registers!
STO 0..9, A..E	Save X in storage register
STO +-x÷ 0..9	Storage register arithmetic: Register OP X → Register Not available for registers A..E!
ST I	Store X in indirect register I
STO (i) STO +-x÷ (i)	Store X in the register indexed by the absolute value of the integer part of I. An error occurs if the I is out of range. Register arithmetic is supported for all values of I. Values of I and associated registers: 0-9=primary R0-R9, 10-19=secondary R0-R9, 20-24=A-E, 25=I
RCL 0..9, A..E	Retrieve value from storage register. Recall register arithmetic is not supported
RC I	Retrieve index register I
RCL (i) *or* (i)	Retrieve register indexed by I
X↔I	Exchange X with index register I
P↔S	Exchange primary and secondary register set values
REG	Displays primary registers R0-R9, A-E & I. For the numbering see STO (i)

Functions (selection)

y^x	Y to the power of X. Y may be negative if X is integer
n!	Faculty of integer argument ≥0
H→H.MS	Convert fractional hours to hours/minutes/seconds in h.mmss format
H←H.MS	Convert hours/minutes/seconds in h.mmss format to fractional hours
H.MS+	Add numbers in hours/minutes/seconds format. To substract add a negative X
DEG	Use trigonometric mode degress (360, default)
RAD	Use trigonometric mode radians (2π)
GRD	Use trigonometric mode grad (400)
D→R	Convert degress (360) to radians (2π)
D←R	Convert radians (2π) to degress (360)
R→P	Convert rectangular coordinates (X,Y) to polar coordinates (r,θ)
R←P	Convert polar coordinates (r,θ) to rectangular coordinates (X,Y)
%	Calculate X percent of Y. Stack does not drop!
%CH	Calculate percentual difference from Y to X. Stack does not drop!
A - E	If the program memory is clear (default after power-on) these keys perform standard operations. If program code has been entered these standard operations must be executed using shifted keys of the number pad
SPACE	Doesn't do anything, only for HP-97 compatibility

Summation

Memory	The summation registers are mapped to the following *secondary* registers: 14=Σx 15=Σx² 16=Σy 17=Σy² 18=Σxy 19=n IMPORTANT: CL REG does *not* clear the summation registers! Use CL REG and then P↔S to exchange the zeroed primary with the secondary register contents!
Σ+	Add X & Y to the summation registers and increment n
Σ-	Substract X & Y from the summation registers and decrement n
STO Σ+	Same as Σ+
RCL Σ+	Retrieve Σx and Σy into X and Y
$\bar{x}$	Calculate mean of X & Y values and place result in X & Y
s	Calculate standard deviation sx & sy and place result in X & Y where: sx=SQRT[$\{n\sum x^2 - (\sum x)^2\} / \{n(n-1)\}$] and similar for sy

Programming

Memory	224 merged program steps
W/PRGM – RUN switch	Selects RUN or PRGM mode
CL PRGM	In PRGM mode: • Clears program memory • Selects FIX 2 • Chooses trigonometric mode degrees (360) • Clears all four flags In RUN mode: Sets program counter to 000
Program editing	• Use SST & BST to step thru the program memory • Delete the currently displayed instruction with DEL (on the CLx key). Also, the program counter moves back one instruction • New instructions are inserted after the currently displayed one
SST	RUN mode: Displays next instruction while SST is held down. Executes instruction when key is released
BST	RUN mode: Displays previous instruction while BST is held down. Backs up program counter to previous instruction when key is released. No code is executed!
LBL 0..9, A..E LBL f a..e	Insert a label. Labels are searched from the current position and it is possible to use the same label multiple times
GTO . nnn	In PRGM and RUN mode sets the program counter to the specified instruction line 0..224
GTO 0..9, A..E GTO f a..e	PRGM mode: Insert jump instruction to given label RUN mode: Sets the program counter to the specified label
GSB 0..9 [GSB] A..E [GSB f] a..e	PRGM mode: Inserts subroutine call to given label. A maximum of 3 subroutine calls can be nested RUN mode: Executes program at the given label Note: When keys A..E or a..e are used it is not necessary to prefix them by GSB, neither in PRGM nor RUN mode
GTO (i)	$I \geq 0$: Jumps to the specified label. Values of I and associated labels: 0..9=LBL 0..9, 10..14=LBL A..E, 15..19=LBL a..e $I < 0$: Jumps back the specified number of program steps The fractional part of I is discarded
GSB (i)	Same as GTO (i) except that a subroutine call is performed
RTN	PRGM mode: Insert return instruction. Returns from subroutine or at the top level halts program and sets program counter to 000 RUN mode: Sets program counter to 000
DSZ	Decrement and skip when zero. Decrements the I register and skips the next instruction if after the decrement the integer part of I is 0 ($-1<I<1$)
DSZ (i)	Same as above except that the storage register indexed by the absolute value of the integer part of I will be decremented. See STO (i) for register numbers
ISZ	Increment and skip when zero. Increments the I register and skips the next instruction if after the

	increment the integer part of I is 0 (-1<I<1)
ISZ (i)	Same as above except that the storage register indexed by the absolute value of the integer part of I will be incremented. See STO (i) for register numbers
SF n, CF n, F? n Flags	There are four flags 0..3. F? n tests a flag: • If flag is set next program instruction is executed • If flag is clear next program instruction is skipped **Flags 2 & 3** are automatically cleared when they are tested using F? **Flag 3** is automatically set when data has been entered or a magnetic card has been read. This is extremely useful in order to determine whether the user wants to enter a variable when pressing A-E (or a-e) or whether he wants to solve an equation for the variable that is associated with the function key. In the former case Flag3 is set in the latter Flag3 is clear.
Relational operators	See calculator for available comparisns between X and Y and Y and 0 • If the relation is true the next program line is executed • If the relation is false the next program line is skipped
-x-	Displays X with a blinking decimal point for 5 seconds. Any key press halts the program
PAUSE	Displays X for about 1 second. If a key is pressed its function is executed and the pause prolonged by another second. Only R/S stops the program execution. It is even possible to read a magnetic card during a PAUSE!

Magnetic Card Reader, Manual Operation

Write program	To write the contents of the program memory to a magnetic card: • Select W/PRGM mode and insert the card from the right side • Program instructions will automatically be written and if necessary, previous contents of the card are erased (as long as the card is not protected by cutting off the "ears") • A maximum of 112 program steps can be written to one side of the card. If there are more instructions present the display shows Crd and the card must be reinserted top-down so that the remaining code can be written in a 2nd pass The following information will be recorded on the magnetic card: • The fact that the card contains program code rather than data • Whether the data corresponds to pass 1 or pass 2 • Whether only one or both sides have been written • State of all four flags • Current trigonometric mode (DEG, RAD, GRD) • Display format (FIX, SCI, ENG and number of digits) • Program data • A checksum
Read program	To read in the program contained on a magnetic card: • Select RUN mode and insert card from the right side • Program instructions will automatically be read and previous contents of the program memory will automatically be overwritten • If necessary Crd is displayed to inform that the 2nd side must be read as

	well. • Note that order of reading the sides doesn't matter! • If an empty card is inserted the program memory is not affected. However, if data has partially been read from the card and an error occurs the current program memory will be lost • Note that all 244 program steps will be affected even if the card contains only a few instructions: The remaining program memory is filled with R/S commands (code 84) • All the information stored on the card (ie. trigonometric mode, flags, etc., see above) is updated • Storage registers and the stack are not affected
W/DATA	Magnetic cards can also contain data (=numbers) instead of program code. To write a data card: • Select RUN mode • Press W/DATA and insert a card when Crd is displayed. All storage register contents will be saved • If the secondary registers are not all 0 then Crd is displayed and the card must be reinserted for a 2nd pass To read a data card: • Select RUN mode • Insert data card. Storage register contents will automatically be read • If necessary Crd is displayed to inform that the 2nd side must be read as well. At this point CLx can be pressed to abort the process in which case the calculator's secondary registers are not affected • Note that order of reading the sides doesn't matter!
MERGE program	Allows to append a program stored on a magnetic card to the program already in memory: • Select RUN mode • Enter GTO . nnn where nnn is the last program line that will *not* be overwritten by the magnetic card data. Or: nnn+1 is the first program line that will receive the code from the card • Press MERGE and insert card. If is Crd displayed re-enter the card top-down for a 2nd pass • Note that no matter how short the merged program is it will always overwrite all the program memory from step nnn+1 to 224
MERGE data	Allows to partially read in storage register data from a data card: • Select RUN mode • Store a number from 0 to 25 in the I register. See STO (i) for register numbers • Press MERGE and insert the data card. If is Crd displayed re-enter the card top-down for a 2nd pass • Storage registers 0 to including the register indexed by I will receive data from the magnetic card. Only the absolute value of the integer part of I is used. If I>25 then all data will be read

Magnetic Card Reader, Programmed Operation

General	Under program control it is possible to: • Read a data card containing storage register values

	• Partially read a data card containing storage register values • Write storage register values to a data card • Read a program card • Partially read (append) instructions from a program card Note that once the program has been started a magnetic card can be inserted in the card reader slot in preparation of the anticipated read or write operation. When the program reaches the PAUSE command (see below) the card reader will sense the presence of the card and read or write it. This allows for unattended operation as long as no 2nd pass is needed to read or write side 2 of the card
Read register values	• The program must contain a PAUSE instruction • When the user enters a magnetic *data* card during the PAUSE the register contents will be read and Flag3 is set. If no card is inserted the program continues normally and Flag3 is clear
Read register values partially	• The program must prepare the I register, see MERGE above • The program must contain a MERGE and then a PAUSE instruction • When the user enters a magnetic *data* card during the PAUSE the register contents will be read starting from register 0 up to including the register specified by I. Flag3 is set. If no card is inserted the program continues normally and Flag3 is clear
Write register values	• The program must contain a W/DATA and then a PAUSE instruction • When a magnetic card is inserted during the PAUSE all register contents will be written. If all secondary registers are 0 no 2nd pass is necessary
Read entire program	• The program must contain a PAUSE instruction • When the user enters a magnetic *program* card during the PAUSE its program instructions will overwrite the entire program memory • Execution will restart at program step 000
Append program	• The program must contain a MERGE and PAUSE instruction • When the user enters a magnetic *program* card during the PAUSE its program instructions will overwrite the program memory starting with the first instruction *after* the PAUSE command • Execution will restart at the instruction following the PAUSE command