HP-16C Quick Reference

General Calculator Control

STATUS	Displays the current number format in the form "C - BB – abcd" C: Indicates the negative number format: 2=two's complement, 1=one's complement, 0=unsigned BB: Indicates the number of bits, can be anything from 1 to 64. Floating point numbers use 56 bits. abcd: Indicates the settings of flags 3, 2, 1 and 0.
WSIZE	If not in floating point mode sets the number of register bits to the value specified in X. Can be anything from 1 to 64. If X is 0 then 64 bits are selected. The stack drops. Changing the word size keeps the current register values if possible. The contents of the storage registers do not change but the storage register boundaries do change! See MEM and RCL
MEM	Displays the number of programs steps P and storage registers r. Registers are automatically converted to program steps. Program space is incremented in junks of 7 bytes. In total the HP-16C contains 203 bytes of memory. Note that the number of storage registers depend on the current number width as set by WSIZE. A storage register always occupies memory in multiples of nibbles (4 bits). Example: If the word size is 16 bits (2 bytes) and progam space is empty then 101 registers are available. If the word size is 64 bits (8 bytes) then there are only 25 registers. To make room for more storage registers program steps must be deleted individually or by using CLEAR PRGM
FLOAT n	Set floating point format where n denotes the number of digits after the decimal point. n can be anything from 0 to 9. "FLOAT ." chooses scientific display mode with exponent. FLOAT automatically chooses 56 bit mode. Also calculates Y • 2^X and stores the result in X. Y, Z, T and Last-X are cleared. Note that the HP-16C uses a BCD floating point format
HEX	Choose hexadecimal number format, indicated by "h". Also, the reverse of the Y • 2^X calculation described for FLOAT is supposed to happen – but that doesn't seem to work
DEC	Choose decimal number format, indicated by "d"
OCT	Choose octal number format, indicated by "o"
BIN	Choose binary number format, indicated by "b"
f HEX	Temporarily displays X in hexadecimal format until the HEX key is released. Works also for DEC, OCT and BIN
< >	If a length number doesn't fit entirely into the display "<" shifts the display one digit to the left and ">" shifts it one digit to the right. Pressing and holding "<" or ">" scrolls the number. Also note that if the number doesn't fit a dot beside the radix indicator hints in what direction the number extends beyond the display. Ie. ".h" indicates that there a more digits beyond the left end of the display.

WINDOW n	The display can be thought of showing only 8 digits from a maximum of 64 possible digits (all 1s in a 64-bit binary number). n selects an 8-digits group from those 64 digits where n=0 corresponds to digits 0 to 7 (=least significant digits), n=1 corresponds to digits 8 to 15 etc. n can be anything from 0 to 7. Be aware that *nothing* is displayed if a window is selected that doesn't contain digits and the display of leading zeros is suppressed. Most – but not all – operations reset the display to window 0
1'S	Choose 1's complement for negative *decimal* numbers, -1=…FE
2'S	Choose 2's complement for negative *decimal* numbers, -1=…FF
UNSIGN	Choose unsigned *decimal* numbers
CHS	Negate number in decimal or FLOAT mode; replace number with its 1's or 2's complement if in HEX, OCT or BIN mode. In unsigned mode replaces X with its 2's complement and sets out-of-range flag 5 (indicated by "G" in the display)
STO n	Store number in register 0 to 9, A to F, .0 to .9 , .A to .F , I (total of 33 registers, I is the index register). To address higher registers the **Indirect Addressing** must be used. Register arithmetic is not available
RCL n	Get value from register. *Important*: When the word size is changed the boundaries of the storage registers change and thus previousely stored values cannot be retrieved correctly any more! However, changing the word size back to the original setting make the old values accessible again. When the word size has been changed then recalling a value from a register in FLOAT mode may or may not cause an error depending on whether the recalled value represents a valid BCD float number.
BSP	Clears the X register or deletes digits during number entry
CLx	Clear the X register even if currently entering digits
CLEAR REG	Clear all storage registers
CLEAR PREFIX	Clear prefix key and in FLOAT mode briefly displays all digits of the mantissa
LST X	Get back last value of X register as it was before the most recent operation
Decimal point	Turn off calculator, press & hold ON, press and hold ".", release ON, release "." to switch between comma and dot for the decimal point
Leading zeros	The display of leading zeros in integer mode is controlled by flag 3. Set flag three (SF 3) to display leading zeros. This does not affect the DEC mode!
Continuous memory	The entire machine state is stored in nonvolatile RAM. Except that the calculator always comes on in RUN mode

Bit Manipulation Functions

See label on the back of the calculator.
Note that bit manipulation functions are not available in FLOAT mode!

SL	Shift one bit left, high bit into carry, insert 0 from right
SR	Shift one bit right, low bit into carry, insert 0 from left
RL	Rotate one bit left, high bit goes into carry and bit0
RR	Rotate one bit right, low bit goes into carry and high bit
RLn	Rotate left X number of bits

RRn	Rotate right X number of bits
MASKL	Create a mask that has the higher X bits set to 1 and store result in X
MASKR	Create a mask that has the lower X bits set to 1 and store result in X
LJ	Adjust X register to the left so that the high bit is set and store in Y. Store the number of bits that X had to be shifted left in X. Example: X=2000h with WSIZE 16 results in X=2 (X had to be shifted left by 2 bits) any Y=8000h (the left-adjusted result)
ASR	Rotate one bit right, low bit into carry and duplicate high bit
RLC	Rotate one bit left, high bit goes thru carry
RRC	Rotate one bit right, low bit goes thru carry
RLCn	Rotate left X number of bits, high bit goes thru carry
RRCn	Rotate right X number of bits, low bit goes thru carry
#B	Count the number of 1-bits in X and overwrite X with the result
RMD	Reminder after division of Y/X
XOR AND OR	Bitwise XOR, AND, OR
NOR	Bit inversion
SB	Set the bit number X, starting from 0
CB	Test the bit number X, starting from 0
B?	Test the bit number 0. Needed for programming
DBLx	Double multiply X • X → (X,Y) where X contains the high and Y the low order bits
DBL÷	Double divide (Y,Z) ÷ X → X where initially Y contains the high and Z the low order bits
DBLR	Double reminder (Y,Z) % X → X where initially Y contains the high and Z the low order bits

Indirect Addressing

General	Note that the index register I is alway 68 bits wide and is never converted to program space. For indirect GTO and GSB as well as ISZ and DSZ see section **Programming**
STO I	Store value in index register. I can hold integer as well a FLOAT numbers. Note that it is not necessary to press the prefix key f when using I or (i) together with STO/RCL
RCL I f I	Retrieve value from index register
STO (i)	Indirect storage: Store X in the register that the I register points to. Values of I and corresponding storage registers: 0…9 → 0…9, 10…15 → A…F, 16…25 → .0….9, 23…31 → .A….F, 32… → not directly accessible registers. Note that register 32 does *not* correspond to the index register I
RCL (i) f (i)	Indirect retrieval
x<>I	Exchange X with the index register. If the value in the index register is larger than the X register can hold then it will be truncated
x<>(i)	Indirect exchange

Programming

P/R	Switch from RUN to Program (PRGM) mode and back. Program line are partially merged. Program code corresponds to the row and column of the corresponding key except for number keys 0-9 where the code directly corresponds to the number. Prefix keys have their separate codes
CLEAR PRGM	RUN mode: Set program counter to 000 PRGM mode: Erase entire program memory
RTN	RUN mode: Return from subroutine or end program and set program counter to 000 PRGM mode: Set program counter to 000
GTO .nnn	Goto program line n in PGM & RUN mode
GTO n	RUN mode: Jump to label n with n=0…9, A…F or I PRGM mode: Set program counter to label n
LBL n	RUN mode: Insert label n with n=0…9, A…F or I Labels are search from the current position downwards. This in mind it is possible - but not advisable – to use a particular label multiple times
R/S	RUN mode: Halt/start program PRGM mode: Enter a halt instruction. It does not set the program counter to 0 when hit
GSB n	RUN mode: Execute program starting at label n with n=0…9, A…F or I PRGM mode: Insert subroutine call to label n. A maximum of 4 GSB/RTNs can be nested.
PSE	Pause program execution for about 1sec and display X
BSP	PRGM mode: Delete current program line and move subsequent lines up
Inserting code	New program lines are inserted *after* the currently displayed line
SST	RUN mode: When held displays the next program line, when released executes the next program line PRGM mode: Steps forward thru program code
BST	RUN mode: Step backwards one program line but do not execute any code PRGM mode: Steps backward thru program code
GTO I GSB I	Indirect GTO and GSB. This cause a jump to or call to the following labels: 0…9 for I=0…9, A…F for I=10…15. If I is outside this range and error occurs. In integer mode the absolute value of I is used, in FLOAT mode the integer part. So by using negativ I values it is *not* possible to jump back a given number of program lines.
Conditional branching	There are various comparisns for the X and Y register as well as bit testing (B?) and flag testing (F?). If the comparisn is not met or the bit or flag not set, the next instruction is skipped. If the comparisn is met or the bit or flag is set, the next instruction is executed
DSZ	Decrement index register and skip next instruction if (after the decrement) index register I is 0. Note that the index register is 68 bits wide; it is not affected by WSIZE.
ISZ	Increment index register and skip next instruction if (after the increment) index register I is 0.

Flags

Flag 0..2	User flags
Flag 3	If set leading zeros are displayed, otherwise they are suppressed
Flag 4	Carry flag, if set a the "C" indicator is activated. On the back of the calculator there's a summary of what operations affect the C and G flag
Flag 5	Out-of-range flag, if set a the "G" indicator is activated
SF n	Set flag, n=0…5
CF n	Clear flag, n=0…5
F?	Test flag, n=0…5. Needed for programming.

Self Tests And Trouble Shooting

Global reset: Clears all continuous memory	Turn off, press & hold ON, press and hold "-", release ON, release "-"
If keyboard does not respond	Press D & ON
If keyboard still does not respond	Remove/reinsert batteries
If keyboard still does not respond	Short installed batteries briefly
Perform selftest and if successful turn on LCD indicators and display: -8,8,8,8,8,8,8,8,8,8,	Turn off, press & hold ON, then press & hold "x", release ON, release "x"
Run above test contiuousely	Turn off, press & hold ON, then press & hold "+", release ON, release "+"
Keyboard test: Press all keys from left to right and top to bottom (start with A). During the test cryptic segment patterns are display. On success "16" is displayed.	Turn off, press & hold ON, then press & hold "÷", release ON, release "÷"