HP-29C Quick Reference

© A. Thimet

Memory

| Permanent memory | • 16 storage registers  
|                  | • X register  
|                  | • 98 program steps  
|                  | • Display format  
| Volatile memory  | • 14 storage registers  
|                  | • Y, Z, T and Last-X registers  
|                  | • Trigonometric mode  
|                  | • Program counter  

Storage

| STO 0-9, .0-.5, i | Save data in one of the 16 permanent storage registers  
| STO ÷x 0-9, .0-.5, i | Register storage arithmetic: Register OP X → Register  
| RCL 0-9, .0-.5, i | Retrieve data from one of the 15 permanent storage registers  
| Indirect addressing | • Volatile registers 16-29 can only be accessed via indirect addressing  
|                   | • R0 contains the index. Only the integer part of the absolute value of R0 will be used  
|                   | • Note that registers .0-.9 correspond to R0=10-15  
|                   | • Press "STO i" or "RCL i" for indirect operations. i is located on the R↓ key and need not be prefixed  
| CLEAR REG | Clears all registers 0-29  

Miscellaneous

| FIX n | Select fixed point format with n digits after the decimal point  
| SCI n | Select exponential format with n valid digits  
| ENG n | Select exponential format where the exponent is always a multiple of 3. Note that with n=0 & 1 routing can occur in front of the decimal point!  
| DEG | Trigonometric mode degrees (360, default)  
| RAD | Trigonometric mode radians (2π)  
| GRD | Trigonometric mode grad (400)  
| y^x | Y to the power of X. Y may be negative if X is integer  
| → H | Convert h.mmss to fractional hours  
| → H.MS | Convert fractional hours to h.mmss format  
| → R | Convert polar coordinates (X=r, Y=θ) to orthogonal coordinates  
| → P | Convert orthogonal coordinates to polar coordinates (X=r, Y=θ)  
| % | Calculate X percent of Y. The stack doesn't drop!  

1
### Summation

**Memory**  
6 summation registers mapped to the top non-volatile storage registers:  
\[ n=R.0 \quad \Sigma x=R.1 \quad \Sigma x^2=R.2 \quad \Sigma y=R.3 \quad \Sigma y^2=R.4 \quad \Sigma xy=R.5 \]

**CLEAR Σ**  
Clear summation registers

**Σ+**  
Add X & Y to the sum registers and increment n

**Σ-**  
Subtract X & Y from the sum registers and decrement n

**x**  
Calculate mean of X & Y values and put result in X & Y register

**s**  
Calculate standard deviation of X & Y values and put result in X & Y register:

\[ sx=\sqrt{\frac{n\sum x^2-(\sum x)^2}{n(n-1)}} \] and similar for sy

### Programming

**Memory**  
98 fully merged program steps  
Program does not halt when it encounters step 00!

**PRGM/RUN**  
Use this switch to select programming or execution mode

**CLEAR PRGM**  
RUN mode: Does nothing  
PRGM mode: Clear all program memory (fills with R/S instructions)

**Program editing**  
- New instructions will be inserted after the currently displayed line
- DEL deletes the currently displayed instruction and displays the previous line

**SST & BST**  
(RUN mode)  
SST: Execute program step-by-step. While the key is held down the next instruction is displayed. When the key is released the instruction is executed  
BST: Same as SST except that no instructions are executed

**SST & BST**  
(PRGM mode)  
Step forward/backward thru program instructions

**LBL 0-9**  
Insert label. The same label can be used multiple times. Labels will be search from the current program counter towards the end of the program memory

**GTO . nn**  
RUN or PRGM mode: Jump to line number nn

**GTO 0-9**  
RUN mode: Set program counter to specified label 0-9  
PRGM mode: Insert jump instruction to label 0-9

**GTO i**  
RUN mode: Set program counter indirectly via R0  
PRGM mode: Insert indirect jump instruction via R0  
Only the integer part of R0 will be used:  
- R0=0..9: Jump to the specified label  
- R0<0: Jump back the given number of instructions in program memory

**GSB 0-9, i**  
RUN mode: Execute program on specified label 0-9 or indirectly via R0  
PRGM mode: Insert subroutine call to label 0-9 or indirect call via R0.  
At most 3 subroutine calls are possible

**RTN**  
RUN mode: Set program counter to 00  
PRGM mode: Insert return from subroutine instruction. At the top level this will halt the program and the program counter will point to the instruction after the RTN

**PAUSE**  
Halt program for about 1 sec and display X register
### HP-29C

<table>
<thead>
<tr>
<th>R/S</th>
<th>RUN mode: Halt program. Note that in order to stop a program during a PAUSE you must hold the R/S key down until the pause is over!</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PRGM mode: Insert halt instruction</td>
</tr>
<tr>
<td>DSZ</td>
<td>Decrements R0 and if the integer part of the result is 0 skips the next program instruction</td>
</tr>
<tr>
<td>ISZ</td>
<td>Increments R0 and if the integer part of the result is 0 skips the next program instruction</td>
</tr>
</tbody>
</table>
| Comparisn | These relational operators are available:  
X<0     X≥0     X≠0     X=0  
X<Y     X≥Y     X≠Y     X=Y  
If the relation is true the next program step is executed.  
If the relation is not true the next program step is skipped. |