Using Menus

A menu redefines the top row of keys by displaying a menu label above each key. If the current menu has more than six labels, ▼▲ is displayed indicating that the [▼] and [▲] keys can be used to display the additional rows of the menu.

Application Menus

BASE  MATRIX  SOLVER  STAT  f(x)

When you select an application menu, all other menus are automatically exited. Within an application, you can select and use any function menu (below).

Function Menus

CATALOG  CLEAR  CONVERT  CUSTOM

Disp  FLAGS  Modes  PGM.FCN

PRINT  PROB  TOP.FCN

Function menus (except for CUSTOM) automatically exit as soon as you press a menu key. To prevent automatic exiting, select the menu twice.
Memory

The Stack
The stack is a workspace for calculations. Each stack register may contain any type of data.

| T | Z | Y | X | Last X |

The Alpha Register

Flags (00-99)

Available Memory
The HP-42S has 8,192 bytes of RAM. After initializing the items in system memory (such as the stack, the Alpha register, and the flags), there’s about 7,200 bytes available for your programs and variables. The storage register matrix (REGS) occupies part of this user memory.

**CATALOG MEM** displays the amount of unused memory. To increase available memory, use the CLP (clear program) and CLV (clear variable) functions to clear items that are no longer needed.

Variables
A variable is a named storage location that may contain any type of data. For example, to store a copy of the X-register into a new variable named ABC, press:

`STO ENTER ABC ENTER`

Variable names can be up to seven characters long.

Note: the variable name REGS is reserved for the storage register matrix (shown on the next page).
When you execute a function that accesses a variable, the calculator automatically displays a menu of existing variable names for you to choose from. For example, to recall the contents of ABC, press:

```
RCL ABC
```

### Storage Registers (REGS)

Each storage register is an element in the matrix REGS.

- **STO** \(nn\) stores a copy of the X-register into register \(nn\).

- **RCL** \(nn\) recalls the contents of a storage register into X.

Initially, there are 25 storage registers; numbered from 00 through 24. Use the SIZE function (in the MODES menu) to change the number of storage registers.

To access registers numbered greater than 99, you must use indirect addressing (see page 7).

Before storing a complex number into a storage register, the entire REGS matrix must be complex.

To make REGS complex, press:

```
0 ENTER [COMPLEX] STO + REGS
```

To convert REGS back to a real matrix, press:

```
RCL REGS [COMPLEX] xy STO REGS
```
Data Types

Real Numbers
Real numbers are the simplest type of data. For example, any number you key into the calculator is a real number.

Complex Numbers
A complex number consists of two real numbers combined to represent a real and imaginary part:

3.16 -i4.12 \((\text{Rectangular coordinate mode})\)

Or, a magnitude and angle:

5.19 \(-52.51\) \((\text{Polar coordinate mode})\)

In Polar mode, complex numbers are automatically normalized so that magnitudes are positive and angles are not larger than 180 degrees.

Complex numbers are entered left-to-right:

\text{left-hand-part [ENTER] right-hand-part [COMPLEX]}

That is, the COMPLEX function converts two real numbers (or matrices) in the X- and Y-registers into a complex number (or matrix). If the X-register contains a complex number (or matrix), the COMPLEX function separates it into its two real components.

Alpha Strings
The Alpha register can hold up to 44 characters. Alpha strings outside the Alpha register are limited to six characters, and can be stored any place a real number can be stored.

Matrices
A matrix can be any size, limited only by the amount of available memory. Each element in a matrix holds a complete number. (See page 12.)
Modes

Angles and Coordinates (MODES):
- DEG: Degrees.
- RAD: Radians.
- GRAD: Grads.
- RECT: Rectangular coordinates.
- POLAR: Polar coordinates.

Other (MODES):
- SIZE: Sets the number of storage registers.
- QUIET: Disables the beeper.
- CPXRES: Complex-result enable.
- REALRES: Real results only.
- KEYASN: Key Assignments; for the CUSTOM menu.
- LCLBL: Local Labels; for the CUSTOM menu.

Display Formats (DISP):
- FIX: Fixed-Decimal.
- SCI: Scientific notation.
- ENG: Engineering notation.
- RDX: Radix Period.
- RDX,: Radix Comma.

Printing (PRINT):
- PRON: Printing On (sets flags 21 and 55).
- PROFF: Printing Off (clears flags 21 and 55).
- MAN: Manual (for printing results).
- NORM: Normal (for printing inputs and results).
- TRACE: Trace (for printing all operations).

Additional modes are described under "Matrix Operations" and "Statistics."

Display Contrast

To darken the display: Press [+] while holding [EXIT].
To lighten the display: Press [−] while holding [EXIT].
Any function or program can be executed with:

```
XEQ ENTER name ENTER
```

where name is a function name or program label. If name is not unique, the global label closest to the permanent end (.END.) has precedence.

If name is a local Alpha label, the calculator searches only the current program. (Local numeric labels in the current program are executed with \( XEQ \) \( nn \)).

**Short Cuts**

**The CUSTOM menu.** CUSTOM has room for 18 assignments. Pressing a menu key in the CUSTOM menu is equivalent to using the XEQ function as described above where the characters assigned to the CUSTOM menu key take the place of name.

**Smart Program Catalog.** The XEQ function automatically displays the program catalog. Specify name by pressing the corresponding menu key.

**Single Stepping.** To execute the next single program instruction (at the current program line), press \( \text{[SST]} \) (or \( \text{[¥]} \) if no menu is displayed).

**The Run/Stop Key.** Pressing \( \text{[R/S]} \) runs the current program (beginning at the current line) or stops a program after the current instruction is complete.

**The Function Catalog.** To display a menu containing all HP-42S functions, press \( \text{[CATALOG]} \) \( \text{[FCN]} \).

**Specifying Function Parameters**

**Numeric Parameters.** Functions that accept numeric parameters prompt you with a cursor for each digit expected. For example, the STO function prompts with \( \text{STO } \_\_ \) and accepts a two-digit register number.
To key in a numeric parameter, simply key in the digits. If you provide a digit for each cursor, the function executes. You can also provide fewer digits and complete the entry with [ENTER].

**Alpha Parameters.** Many functions that accept numeric parameters also accept Alpha parameters. Often, the parameter you want is an object that already exists, so the calculator displays a menu for quick entry. If the item does not exist, use the ALPHA menu to type it. For example, to create a variable:

[**STO**] [ENTER] SONJA [ENTER]

**Stack Parameters.** Any function that accepts a storage register as a parameter also accepts a stack register. To specify a stack register, press the decimal key and then a menu key for the stack register. For example, to recall a copy of the Z-register:

[**RCL**] [ST] [Z]

**Indirect Addressing.** Rather than providing an actual parameter, you can specify the variable or register that contains the parameter. To do this, use the same menu as for stack parameters. For example, to display the contents of the variable or register named in R12:

[**PGM.FCN**] [VIEW] [IND] [12]

You can also use stack registers with indirect addressing. For example, to clear the variable whose name is in the Y-register:

[**CLEAR**] [CLV] [ST] [Y]

Notice that **IND** is not needed because the CLV function takes only Alpha parameters (variable names).
Programming

Program-Entry

- **PRGM** toggles in or out of Program-entry mode.
- **GTO** moves to a new program space.
- **GTO** moves to line number \( nnnn \).
- **SST** deletes the current program line.
- **SST** moves to the next program line.*
- **BST** moves to the previous program line.*

(* Use \( \downarrow \) or \( \uparrow \) if no menu is displayed.)

Labels

A program label is simply a marker used to identify a program or a routine within a program.

**Global labels** can be accessed from anywhere in memory (and therefore should be unique). Global labels are distinguished from local labels with quotation marks (such as \( \text{LBL} \ "\text{SAMPLE}" \)).

**Local labels** can be accessed only within the current program (and should be unique within the current program). There are two types of local labels:

- Numeric (\( \text{LBL} \ 00 - \text{LBL} \ 99 \))
- Alpha (\( \text{LBL} \ A - \text{LBL} \ J \) and \( \text{LBL} \ a - \text{LBL} \ e \))

The Do-If-True Rule

The do-if-true rule determines how program lines are executed when a conditional function is encountered. If the condition is "true," the line immediately following the conditional is executed. If the condition is "false," the line following the conditional is skipped.

Looping

The ISG and DSE functions control looping. Each accesses a variable or register containing a control
number in the form $ccccccc.fffii$; where $ccccccc$ is the current counter value, $fff$ is the final counter value, and $ii$ is the increment size (default is 1). Both ISG and DSE follow a variation of the do-if-true rule: if the count is not complete, the line following the instruction is executed (usually a branch to the top of the loop). For example, this program segment counts from 1 to 52 by threes (executing the loop 18 times) and then beeps.

**Using a Variable Menu**

A variable menu may be displayed by the Solver or Integration applications, or by the VARMENU function within a program. Each label in the menu represents a variable. While the menu is displayed, you can:

**Store a value into a variable:**
Key in the value and then press the menu key.

**Recall the contents of a variable:**
Press \[ \text{RCL} \] and then the menu key.

**View the contents of a variable without recalling it:**
Press \[ \text{M} \text{(shift)} \] and then hold the menu key down.

**Select a variable:**
Press the menu key without keying in a number first. This action places the variable name in the Alpha register and continues execution.

(For the Solver, this is how you select the unknown variable. For Integration, this is how you select the variable of integration.)

You can select and use any function menu without exiting from the variable menu.
The Solver

The Solver is a root finder that allows you to solve for an unknown variable in an expression, given values for all the other variables. Expressions are written as programs. There are three parts to a Solver program:

- The program must begin with a **global label**.
- Immediately following the global label, **menu variables** are declared with MVAR instructions.
- Finally, the body of the program should **evaluate the expression**. Recall the variables as they are needed and calculate \( f(x) = 0 \) for your expression of one or many variables.

After entering the program, these are the steps for using the Solver:

1. Press \([\text{SOLVER}]\).
2. Select a Solver program from the menu.
3. Use the variable menu to store a value into each of the known variables. Optional: store one or two guesses into the unknown variable to direct the Solver to a solution.
4. Solve for the unknown variable by pressing the corresponding menu key.

**A Simple Example:** For the expression \( A + B = C \), rewrite the expression as \( A + B - C = 0 \). The Solver program looks like this:

```
01 LBL "SIMPLE"   05 RCL "A"
02 MVAR "A"   06 RCL+ "B"
03 MVAR "B"   07 RCL- "C"
04 MVAR "C"   08 END
```

Hint: create the variables before entering the program.
After entering the program, you can use it to solve for any variable, given a value for each of the others. For example, find $A$ when $B = 12$ and $C = \log(B)$.

Select the program: 

```
Store B: 12  E
Store C:  \[TOP.FCN\] LOG  C
Solve for A:  A
```

The TOP.FCN menu is used to execute LOG (one of the top-row functions) without exiting from the Solver.

**Numeric Integration**

The Numeric Integration application allows you to calculate an approximation of a definite integral. The integrand, $f(x)$, is written as a program similar to a Solver program (see the previous page). That is, the program must use a global label, declare the menu variables, and evaluate $f(x)$.

After entering the integrand program, here are the steps for using the Integration application:

1. Press $\int f(x)$.
2. Select an integrand program from the menu.
3. Use the variable menu to store a value into each of the variables that should remain constant.
4. Select the variable of integration by pressing the corresponding menu key.
5. Store the lower limit ($LLIM$), the upper limit ($ULIM$), and the accuracy factor ($ACC$).
6. Press $\int$ to calculate the integral. The approximation for the integral is returned to the X-register and the uncertainty of computation is returned to the Y-register.
Matrix Operations

To create a new $m \times n$ matrix, enter the dimensions:

$m \ [\text{ENTER}] \ n$ (for $m$ rows and $n$ columns)

and then press:

- **[MATRIX] NEW** for a matrix in the X-register,
- **[MATRIX] ▼ DIM [ENTER] name [ENTER]** for a matrix in a variable. If the matrix already exists, the DIM function redimensions it.

To edit the matrix in the X-register:

- **[MATRIX] EDIT**

To edit a named matrix:

- **[MATRIX] ▼ EDITH name**

When a matrix is being edited it is said to be indexed. (To index a named matrix without editing it, use the INDEX function.) Whenever there's an indexed matrix, two pointers are used to indicate the row and column of the current element: $l$ and $J$, respectively.

**Wrap and Grow Modes.** If the index pointers are positioned to the last (lower-right) element in a matrix and you move to the right one position:

- The pointers wrap around to the first element of the matrix (Wrap mode).
- Or, the matrix grows by one complete row and the pointers move to the new row (Grow mode).

Wrap mode is automatically selected whenever you enter or exit the Matrix Editor. (The WRAP and GROW functions are in the second row of the Editor menu.)

**Matrix Arithmetic.** Most arithmetic and other operations work for matrices just as for individual numbers. Anytime a matrix is used in a mathematical operation with a complex number, the result is a complex matrix.
Therefore, you can make any matrix complex by adding 0 + i0 to it:

\[
0 \text{ [ENTER]} \text{(COMPLEX)} +
\]

or

\[
0 \text{ [ENTER]} \text{(COMPLEX)} \text{ STO + name}
\]

To solve a system of simultaneous linear equations represented by the matrix equation \( AX = B \):

1. Press \( \text{[MATRIX]} \text{ SIMQ} \).
2. Key in the number of unknowns. The calculator automatically creates or redimensions the matrix variables \( MATA \), \( MATB \), and \( MATX \).
3. Optional: If your equations involve complex numbers, make \( MATA \) and/or \( MATB \) complex (as shown at the top of this page).
4. Press \( \text{MATA} \); fill the matrix; press \( \text{EXIT} \).
5. Press \( \text{MATB} \); fill the matrix; press \( \text{EXIT} \).
6. Press \( \text{MATX} \) to calculate the solution matrix. Use the Matrix Editor keys to view the results.

Statistics

Statistical data is accumulated into 6 or 13 sequential storage registers (see page 3). Initially, the first summation register is \( R_{11} \). Use the \( \Sigma \text{REG} \) function to change the location of the first summation register. \( \Sigma \text{REG} \) does not move the data in the registers.

First, set the appropriate summation mode:

\[
\text{STAT} \downarrow \text{ALL } \Sigma
\]

or

\[
\text{STAT} \downarrow \text{LINE } \Sigma
\]

Next, clear the summation registers:

\[
\text{CLEAR} \text{ CL } \Sigma
\]
Then, accumulate the data:

For each x-y data pair: y-value \( \text{ENTER} \) x-value \( \Sigma^+ \)
or For each single-point data value: x-value \( \Sigma^+ \)
or For x-y data pairs stored in a two-column matrix (x-values in column 1; y-values in column 2): Place the matrix in the X-register and then press \( \Sigma^+ \).

To undo mistakes:

Put the incorrect data in the stack (try \( \text{LAST} \).x). Press \( \Sigma^- \).
Continue accumulating data.

To select a curve model for forecasting:

Press \( \text{STAT} \ CFIT \ MODL \) and then one of the following:

- LINF linear model: \( y = mx + b \)
- LOGF logarithmic model: \( y = m \ln(x) + b \)
- EXPF exponential model: \( \ln(y) = mx + \ln(b) \)
- PWRF power model: \( \ln(y) = m \ln(x) + \ln(b) \)
- BEST selects the model that returns the best correlation coefficient.

Base Conversions

Real numbers are displayed according to the current base mode (Hexadecimal, Decimal, Octal, or Binary). You can change the base mode using the BASE menu or by manually executing HEXM, DECM, OCTM, or BINM. Decimal mode is automatically selected when you exit from the BASE menu.

Press and hold \( \text{SHOW} \) to display:

- A hexadecimal, decimal, or octal number in full-precision decimal form.
- Or, all 36 bits of a binary number.
When the BASE menu is displayed, the following keys are temporarily redefined with these integer functions:

+/-    BASE+ /-  36-bit 2’s complement.
÷      BASE÷    36-bit integer divide.
×      BASE×    36-bit integer multiply.
-      BASE-    36-bit integer subtract.
+      BASE+    36-bit integer add.

Bits are numbered from right to left beginning with 0. Bit 35 (the most significant bit) is the sign bit. Negative numbers are represented in 2’s complement form. Nondecimal numbers longer than 36 bits are displayed as < Too Big >.

**HP-42S Functions**

- **ABS** Absolute value.
- **ACOS** Arc cosine.
- **ACOSH** Arc hyperbolic cosine.
- **ADV** Advance paper.
- **AGRAPH** Alpha graphics.
- **AIP** Alpha integer part.
- **ALENG** Alpha length.
- **ALL** All display format.
- **ALLΣ** AllΣ mode (13 summation registers).
- **AND** Logical AND.
- **AOFF** Alpha off.
- **AON** Alpha on.
- **ARCL** Alpha recall.
- **AROT** Alpha rotate.
- **ASHF** Alpha shift.
- **ASIN** Arc sine.
- **ASINH** Arc hyperbolic sine.
- **ASSIGN** Assign CUSTOM menu key.
- **ASTO** Alpha store.
- **ATAN** Arc tangent.
- **ATANH** Arc hyperbolic tangent.
- **ATOX** Alpha to X.
- **AVIEW** Alpha view.
- **BASE+** Base add.
- **BASE−** Base subtract.
- **BASE×** Base multiply.
- **BASE÷** Base divide.
- **BASE+ /−** Base change sign (2’s complement).
- **BEEP** Beep.
- **BEST** Best fit model.
- **BINM** Binary mode.
- **BIT?** Bit test (x<sup>th</sup> bit of y).
- **BST** Back step.
- **CF** Clear flag.
- **CLA** Clear Alpha register.
- **CLALL** Clear all memory.
- **CLD** Clear display.
- **CLKEYS** Clear CUSTOM menu keys.
- **CLLCD** Clear LCD.
CLMENU Clear the programmable MENU.
CLP Clear program.
CLRG Clear registers.
CLST Clear stack.
CLV Clear variable.
CLX Clear X-register.
CLΣ Clear summation registers.
COMB Combinations.
COMPLEX Complex.
CORR Correlation.
COS Cosine.
COSH Hyperbolic cosine.
CPXRES Complex-result enable.
CPX? Complex test.
CROSS Cross product.
CUSTOM CUSTOM menu.
DECM Decimal mode.
DEG Degrees mode.
DEL Delete program lines.
DELAY Printer delay time.
DELR Delete matrix row.
DET Determinant.
DIM Dimension matrix.
DOT Dot product.
DSE Decrement, skip if less than or equal to zero.
EDIT Edit matrix in X-register.
EDITN Edit named matrix.
END End of a program.
ENG Engineering display format.
ENTER Enter.
EXITALL Exit all menus.
EXPF Exponential fit model.
E↑X e^x.
E↑X-1 e^x-1.
FC? Flag clear test.
FC?C Flag clear test, clear.
FCSTX Forecast x-value.
FCSTY Forecast y-value.
FIX Fixed-decimal display format.
FNRM Frobenius norm.
FP Fractional part.
FS? Flag set test.
FS?C Flag set test, clear.
GAMMA Gamma.
GETKEY Get key code.
GETM Get matrix.
GRAD Grads mode.
GROW Grow mode.
GTO Go to.
HEXM Hexadecimal mode.
HMS+ Hours-minutes-second add.
HMS- Hours-minutes-seconds subtract.
I+ I increment (next row).
I- I decrement (prev row).
INDEX Index matrix.
INPUT Input.
INSR Insert row.
INTEG Integrate.
INVRT Invert matrix.
IP Integer part.
ISG Increment, skip if greater.
J+ J increment (next column).
J- J decrement (previous column).
KEYASN  Key-assignments
 mode.
KEYG    On key, go to.
KEYX    On key, execute.
LASTX   Last x.
LBL     Label.
LCLBL   Local label mode.
LINF    Linear fit model.
LINΣ    Linear mode (six
         summation registers).
LIST    List program lines.
LN      Natural logarithm.
LN1+x   Natural logarithm for values close
to zero.
LOG     Common logarithm.
LOGF    Logarithmic fit.
MAN     Manual printing.
MAT?    Matrix test.
MEAN    Mean (average).
MENU    Programmable
         MENU.
MOD     Modulo.
MVAR    Menu variable.
N!       Factorial.
NEWMAT  New matrix.
NORM    Normal printing.
NOT     Logical NOT.
OCTM    Octal mode.
OFF     Off.
OLD     Old element value.
ON      Continuous on.
OR      Logical OR.
PERM    Permutations.
PGMINT  Program to
         integrate.
PGMSLV  Program to
         solve.
PI      PI.
PIXEL   Pixel on.
POLAR   Polar mode.
POSA    Position in Alpha.
PRA     Print Alpha.
PRLCD   Print LCD.
PROFF   Printing off.
PROMPT  Prompt.
PRON    Printing on.
PRP     Print program.
PRSTK   Print stack.
PRUSR   Print user
         (variables and labels).
PRV     Print variable.
PRX     Print X-register.
PRΣ     Print summation
         registers.
PSE     Pause.
PUTM    Put matrix.
PWRF    Power fit.
QUIET   Quiet mode.
RAD     Radians mode.
RAN     Random number.
RCL     Recall.
RCL+    Recall add.
RCL-    Recall subtract.
RCLx    Recall multiply.
RCL÷    Recall divide.
RCLEL   Recall element.
RCLIJ   Recall IJ pointers.
RDX,    Radix comma.
RDX.    Radix period.
REALRES Real-results
         only.
REAL?   Real test.
RECT    Rectangular mode.
RND     Round.
RNRM    Row norm.
ROTXY   Rotate y by x bits.
RSUM    Row sum.
RTN     Return.
R< >R   Row swap row.
R↑      Roll up.
R↓      Roll down.
SCI     Scientific notation.
SDEV Standard deviation.
SEED Seed (for RAN).
SF Set flag.
SIGN Sign.
SIN Sine.
SINH Hyperbolic sine.
SIZE Size of REGS.
SLOPE Slope.
SOLVE Solve for variable.
SQRT Square root.
SST Single step.
STO Store.
STO+ Store add.
STO- Store subtract.
STO× Store multiply.
STO÷ Store divide.
STOEL Store element.
STOI) StoreIf pointers.
STOP Stop program.
STR? String test.
SUM Ix and Iy.
TAN Tangent.
TANH Hyperbolic tangent.
TONE Tone (0-9).
TRACE Trace printing.
TRANS Transpose.
UVEC Unit vector.
VARMENU Variable menu.
VIEW View.
WMEAN Weighted mean.
WRAP Wrap mode.
X< > x exchange.
X< >Y x exchange y.
Test functions:

\[
\begin{align*}
X<0? & \quad X<Y? \\
X\leq0? & \quad X\leq Y? \\
X=0? & \quad X=Y? \\
X\neq0? & \quad X\neq Y? \\
X\geq0? & \quad X\geq Y? \\
X>0? & \quad X> Y? \\
\end{align*}
\]

Note: If you execute an HP-41 function, it is automatically converted into the corresponding HP-42S function.
Using the ALPHA Menu

To type an Alpha string into the Alpha register:

1. Press \[ALPHA\] to select the ALPHA menu.
2. Optional: press \[ENTER\] to turn on the cursor (in Program-entry mode, inserts the ± symbol).
3. Type the string using the characters shown below. Use \[ (shift) to type lowercase letters.
4. Press \[EXIT\] or \[ENTER\].

Also see "Alpha Parameters" on page 7.

Characters in the ALPHA Menu

<table>
<thead>
<tr>
<th>ALPHA Menu</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>ÄÄ</td>
<td>Ä</td>
<td>Ä</td>
<td>Ä</td>
<td>Ä</td>
<td>Ä</td>
</tr>
<tr>
<td>FGHI</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>JKLM</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>NOPQ</td>
<td>N</td>
<td>O</td>
<td>P</td>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>ÅÖ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSTUV</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
</tr>
<tr>
<td>WXYZ</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>▼</td>
<td>▲</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{</td>
<td>{</td>
<td>{</td>
<td>{</td>
<td>{</td>
<td></td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
<td>&lt;</td>
<td>&lt;</td>
<td>&lt;</td>
<td></td>
</tr>
<tr>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>MATH</td>
<td>Σ</td>
<td>∫</td>
<td>∏</td>
<td>^</td>
<td>v</td>
</tr>
<tr>
<td>PUNC</td>
<td>,</td>
<td>;</td>
<td>:</td>
<td>!</td>
<td>?</td>
</tr>
<tr>
<td>MISC</td>
<td>$</td>
<td>*</td>
<td>#</td>
<td>/</td>
<td></td>
</tr>
</tbody>
</table>

You can also use the following keys to type characters: \[%, π, E, +, -, ×, ÷, :, and 0 - 9\]
## Flags

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-10</td>
<td>User Flags</td>
</tr>
<tr>
<td>11</td>
<td>Auto Execute</td>
</tr>
<tr>
<td>12</td>
<td>Print Double-wide</td>
</tr>
<tr>
<td>13</td>
<td>Print Lowercase</td>
</tr>
<tr>
<td>15-16</td>
<td>Print Mode</td>
</tr>
<tr>
<td>19-20</td>
<td>General Use</td>
</tr>
<tr>
<td>21</td>
<td>Printer Enable</td>
</tr>
<tr>
<td>22</td>
<td>Numeric Input</td>
</tr>
<tr>
<td>23</td>
<td>Alpha Input</td>
</tr>
<tr>
<td>24</td>
<td>Ignore Range Errors</td>
</tr>
<tr>
<td>25</td>
<td>Ignore Next Error</td>
</tr>
<tr>
<td>26</td>
<td>Beeper Enable</td>
</tr>
<tr>
<td>27</td>
<td>CUSTOM Menu</td>
</tr>
<tr>
<td>28</td>
<td>Radix Mark Period</td>
</tr>
<tr>
<td>29</td>
<td>Digit Separators</td>
</tr>
<tr>
<td>30</td>
<td>Stack Lift Disable</td>
</tr>
<tr>
<td>34-35</td>
<td>AGRAPH Control</td>
</tr>
<tr>
<td>36-39</td>
<td>Number of Digits</td>
</tr>
<tr>
<td>40-41</td>
<td>Display Format</td>
</tr>
<tr>
<td>42</td>
<td>Grads Mode</td>
</tr>
<tr>
<td>43</td>
<td>Radians Mode</td>
</tr>
<tr>
<td>44</td>
<td>Continuous On</td>
</tr>
<tr>
<td>45</td>
<td>Solving</td>
</tr>
<tr>
<td>46</td>
<td>Integrating</td>
</tr>
<tr>
<td>47</td>
<td>Variable Menu</td>
</tr>
<tr>
<td>48</td>
<td>Alpha Mode</td>
</tr>
<tr>
<td>49</td>
<td>Low Battery Power</td>
</tr>
<tr>
<td>50</td>
<td>Message</td>
</tr>
<tr>
<td>51</td>
<td>Two-Line Message</td>
</tr>
<tr>
<td>52</td>
<td>Program-Entry Mode</td>
</tr>
<tr>
<td>53</td>
<td>INPUT</td>
</tr>
<tr>
<td>55</td>
<td>Printer Existence</td>
</tr>
<tr>
<td>56</td>
<td>Linear Model</td>
</tr>
<tr>
<td>57</td>
<td>Logarithmic Model</td>
</tr>
<tr>
<td>58</td>
<td>Exponential Model</td>
</tr>
<tr>
<td>59</td>
<td>Power Model</td>
</tr>
<tr>
<td>60</td>
<td>AllΣ Mode</td>
</tr>
<tr>
<td>61</td>
<td>Log Model Invalid</td>
</tr>
<tr>
<td>62</td>
<td>Exp Model Invalid</td>
</tr>
<tr>
<td>63</td>
<td>Pwr Model Invalid</td>
</tr>
<tr>
<td>65</td>
<td>Matrix Editor In Use</td>
</tr>
<tr>
<td>66</td>
<td>Grow Mode</td>
</tr>
<tr>
<td>68-71</td>
<td>Base Mode</td>
</tr>
<tr>
<td>72</td>
<td>Local-Label Mode</td>
</tr>
<tr>
<td>73</td>
<td>Polar Mode</td>
</tr>
<tr>
<td>74</td>
<td>Real-Result Only</td>
</tr>
<tr>
<td>75</td>
<td>MENU</td>
</tr>
<tr>
<td>76</td>
<td>Edge Wrap</td>
</tr>
<tr>
<td>77</td>
<td>End Wrap</td>
</tr>
<tr>
<td>81-99</td>
<td>User Flags</td>
</tr>
</tbody>
</table>

Flags 36-80 cannot be altered with SF, CF, FS?C, or FC?C.

---

**The HP-42S Quick Reference Guide**

by

**TWENTYEIGHT STREET PUBLISHING**

912 NW 28th Street, Corvallis, Oregon 97330-4428

© Copyright 1988 Dex Smith. All rights reserved.

HP Part No. 92222E Printed in U.S.A. October 1988