program. Similarly, if your program begins with a label (A *thru* E), and ends with a RTN you must start it with the program control key identified by the label.

Programs controlled by R/S should be at the top of memory so that they can be easily accessed by pressing RTN R/S.

> **Note:** Generally R/S should not be used to start a program beginning with a label.

Writing a Program Controlled by R/S. Switch to W/PRGM mode and press f PRGM. Then key in the following program which calculates $\sqrt{x^2 + y^2}$.

Keys	Comments	Keys	Comments
ENTER↑	Save copy of x just keyed in.	f^{-1}	Then calculate x^2.
R/S	Stop to key in y.	√x	
f^{-1}	Then calculate y^2.	+	
√x		f	
g x⇄y		√x	
		R/S	Stop the program.

Notice, in particular, that the program does not begin with LBL or end with RTN.

To run the program switch to RUN mode and press RTN to move the program pointer to the top of memory. Then key in a value for x and press R/S. When the program stops again, key in a value for y and again press R/S. The program then stops again to display the answer. Now switch to RUN mode and try the following example.

Example. Calculate $\sqrt{7^2 + 9^2}$.

Press	See Displayed	
RTN	0.00	
7 R/S	7.00	
9 R/S	11.40	The answer.

Program Stops

Blinking Display Stops. Errors that cause a blinking zero display, if executed in a program, also stop the program. Stop the blinking by pressing any key (CLX *is recommended*). You can then identify the reason for the stop by switching momentarily to W/PRGM mode to see the code of the offending operation. You can also use g LSTX to recover the last value in the display.

Normal Stops. To confirm that a program stops normally (i.e., *via a* RTN *or* R/S), switch momentarily to W/PRGM mode and observe the displayed code. It should be 24 or 84.

Accidental Stops. Remember, pressing any key will stop a program. Be careful to avoid pressing keys during program operation.

Cued Stops. If memory permits, it is sometimes helpful to put a familiar number into the X-register before stopping for data. Thus when the program stops, the displayed number identifies the desired input. For example, if your program requires eight stops for input, it is helpful to have the numbers 1 thru 8 appear so you know which input is needed.

If a cue number is created as a program step immediately preceding the R/S, it is not lifted into the stack and the number is overwritten by the data you key in. Cue numbers generated by other means (*recalled from a register, or calculated*) will be lifted.

Overflow Stops. If, during the course of a calculation, you exceed the dynamic range of the machine, a running program will be halted. The display will show 9.999999999 99.

Underflow Stops. If, during the course of a calculation, you calculate a number that is too small in magnitude ($<10^{-99}$) to be carried in a register, the register is set to zero and the program stops, if running.

Writing Programs to Solve Your Problems

In reading this manual, we hope that you have learned from the text and example programs how to program your HP-65. But you may be asking yourself: How do I write a program to solve *my* problem? It is the purpose of this section to briefly describe one approach you might use.