

Switch to W/PRGM mode and press **f** **PRGM**. Now key in the following list of keys:

Keys	Comments	Keys	Comments
LBL	Beginning of program.	R/S	Stop to key in price.
A		x	Quantity \times price.
f	Initialize routine.	.	Calculate discounted price.
STK		8	
LBL	Identify place to start repetition.	5	
3		x	Add to previous total.
R/S	Stop to key in quantity.	+	
ENTER	Copy quantity to Y.	GTO	Repeat, starting at label 3.
		3	

Notice in particular that there is no **RTN** needed at the end of this program. This is because the program is a never-ending loop. And it already stops each time through the loop to let you key in new data.

When the program stops the first time, you key in the quantity of the item and press **R/S** to start the program running again. **R/S** always starts a halted program at the current position of the activated pointer. When the program stops again, you key in the price of the item and again press **R/S**. It calculates the total and returns to label 3 where it stops to receive the next quantity. A running total is displayed. Switch back to RUN mode and try it now.

Example. Assume that you get a 15% discount on the following purchases:

Quantity	Price of Each
5	\$2.00
7	\$4.00
8	\$5.00
22	\$6.00

Calculate the cumulative cost.

Press	See Displayed	
A	0.00	The stack is cleared.
5 R/S	5.00	The first quantity.
2 R/S	8.50	Running total.
7 R/S	7.00	
4 R/S	32.30	Running total.
8 R/S	8.00	
5 R/S	66.30	Running total.
22 R/S	22.00	
6 R/S	178.50	Cumulative cost.

If a **R/S** in a program is immediately preceded by a numerical entry from the program, that number will be overwritten by an entry from the keyboard. This feature allows a program to display prompting information that will not be lifted in the stack. Except for this case, **R/S** does not affect the stack lift.

Note: Digits occurring as program steps immediately following a **R/S** should be separated from the **R/S** by an **ENTER** *.

Controlling Your Program with R/S. Up to this point, each program you have written has begun with a label and ended with a return. We have taught you to program this way because it was judged to be the most convenient for most people and the most used in practice. However, the great versatility of the HP-65 does not confine you to one method. **R/S** can be used to advantage to run initialization routines and even whole programs without using labels and saving valuable memory steps in the process.

The rule for using **R/S** is simple: Pair a **R/S** with a **R/S**. In other words, if you plan to initiate execution of your program with **R/S**, a **R/S** must be used as a program step to halt the

* The reasons for this are discussed on p. 100 under **SST** Execution.