The following program was written using the third approach. Both a and b values are stored before c is calculated. Switch to W/PRGM mode, press f PRGM and key in the following list of keys to see how this program works.

Keys	Comments	Keys	Comments
LBL	Beginning of a storage routine.	RCL 1	Recall a.
A		ENTER↑	Calculate a^2.
STO 1	Store a.	×	
RTN	Stop.	RCL 2	Recall b.
LBL	Beginning of b storage routine.	ENTER↑	Calculate b^2.
B		×	
STO 2	Store b.	+	$a^2 + b^2$
RTN	Stop.	f	
LBL	Beginning of c calculation.	√x	The answer.
C		RTN	

Now switch to RUN mode and see if the program works.

Example. Calculate c for $a = 10$ and $b = 5$. For $a = 78$ and $b = 22$. For $a = 78$ and $b = 10$.

Press	See Displayed	
10 A	10.00	Key in a value.
5 B	5.00	Key in b value.
C	11.18	The answer.
78 A	78.00	Key in a value.
22 B	22.00	Key in b value.
C	81.04	The answer.
10 B	10.00	Key in new b value only.
C	78.64	The answer.

Switch again to W/PRGM mode and record the program on an unprotected magnetic card. Then mark the card as you had originally planned.

Flowcharting

One way to help you with step 2 (*deciding how the problem is to be solved*) is by means of a "flowchart." Flowcharts logically pictorialize the solution to a programming task. They are sometimes drawn long before the actual keystrokes are figured out. While flowcharting your problem, you might change or simplify your approach, see a flaw in your logic, etc. After several attempts (*even for experienced programmers*) you should have a workable flowchart and, once you do, your programming task is greatly reduced.

Any flowchart that you draw is useful, but a few basic flowcharting conventions are described briefly here. Terminal (*that is, starting or ending*) activities are represented by ovals. Arrows indicate the flow of operations between the terminals. Most calculator operations are represented by rectangles. A diamond represents a decision point. If the information within the diamond is computed as *"YES"* (*the condition is met*), the flow continues sequentially; if it is computed as *"NO"* (*the condition is not met*), the flow continues after skipping two steps.

The flowchart for our simple program is shown on the following page. As you can see, once the flowchart is finalized, the program can be written relatively easily.

A complete discussion of flowcharting isn't possible here. It would take many volumes. If you want to learn more about it, you should consult a reference devoted to the subject.

Every program that you write, even the simplest, is written using our three steps, though you may find that with practice you can do much of the work in your head.

1. You must first define your problem.
2. Then you must decide how it is to be solved.
3. And finally you must write out the steps that the calculator will use to solve it.

Good luck!