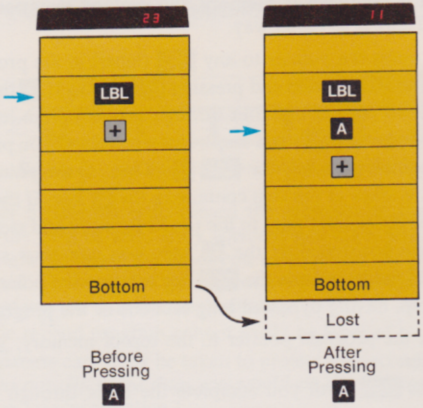
As you can see in the drawing below, when a step is inserted, the bottom step of memory is lost.



Do not concern yourself with the bottom step of memory when inserting unless the display indicates that memory is full.

Insert operations are not performed for the second key of a merged code since the second keystroke uses the same memory location as the first.

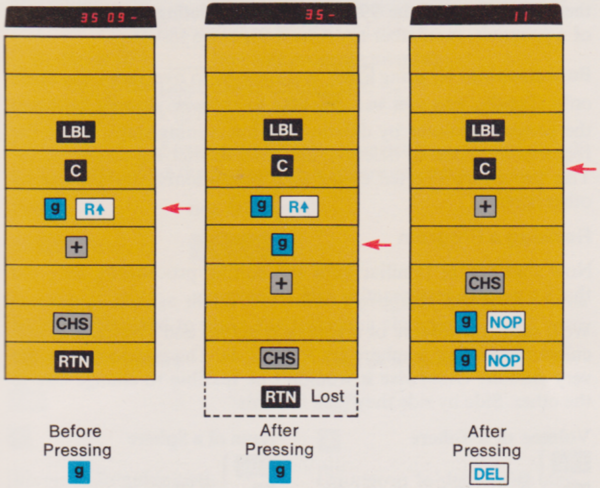
When the program pointer is at the bottom of memory, insert operations are not performed. If the pointer is at the bottom, and you try to insert a step, the code(s) will be generated in the display, but will not go into memory.

Delete Operation

Deleting steps in a program is easily accomplished by following the procedure below:

1. Position the program pointer to display the code of the instruction to be deleted.
2. Press g DEL (delete) in W/PRGM mode. The instruction is removed and the program pointer moves up to the previous step in memory.

The drawings below will help you to understand what happens in program memory when you delete a step.



As seen above, the g key is first interpreted as part of a program operation and it is inserted into memory, pushing down all steps below it. Since memory is full, the bottom step of memory is therefore lost, as in any insert operation. When the suffix key DEL is keyed in, the delete operation is recognized by the calculator and both the g key and the incorrect step are deleted.

The program pointer moves up two steps and g NOP's fill the two vacant steps at the bottom of memory. You will, of course, want to reinsert the last step when this happens. (*For programs shorter than 100 steps—no dashes in the display—no concern need be given the bottom of memory.*)

Deleting Consecutive Steps. To delete a sequence of program steps, position the program at the last step in the sequence. Each time g DEL is pressed, the pointer will backstep to display the next step to be deleted.