LIFO (Last In First Out) Stack is one of the most important concepts in Computer Science, so much so that the computer is design around that concept. The CPU (Central Processing Unit) has a register SP, known as the Stack Pointer, to work specifically with the stack operations.

The most important operations for a Stack are the PUSH and the POP instructions. The Stack is generally based on an array of memory elements, say S(10). There is an integer index to be used as the Stack Pointer.

Assume the Stack Pointer, SP, is pointing to element S(4) at the present time, the PUSH X instruction would increment SP from 4 to 5 and then place the value of X into S(5). The Stack Pointer is now set to a "higher" stack. Another instruction, PUSH Y, would cause SP to become 6, etc. The POP Z instruction would take S(SP) and moved it to variable Z and then decrement SP. The moved value is still in the array S but it will be erase the next time a number is place on the stack by a PUSH instruction.

The Classic HP Calculator is based on a Stack and the method for use is known as the Reversed Polish Notation (RPN). There is no need for parentheses (...) or many levels of parentheses when evaluating an arithmetic expression. The ENTER key is the PUSH instruction to put a value onto the stack and the POP instructions are executed when you do arithmetic operations such as ADD, MULTIPLY, etc.

Whenever you type a number on the calculator, it is recorded on the top of the stack. When press ENTER, it actually provided room for another number on stack. The previous number is actually copied onto this new location with intention that the newly entered number would replace it.

After several numbers have been entered onto the stack, the arithmetic instruction would POP the first two numbers off the stack and perform the arithmetic operation and then PUSH the result back onto the stack. For example, the ADD instruction would take elements, S(SP) and S(SP-1) and add them and then PUSH the sum back onto the Stack. So the operation would do SP(SP-1)=S(SP-1)+S(SP) and after the operation, SP would take on the value of SP-1 because the operation popped two objects and pushed back only one.

For operations such as SQRT, the operation will simply be S(SP)=sqrt(S(SP)), the Stack Pointer would not change.