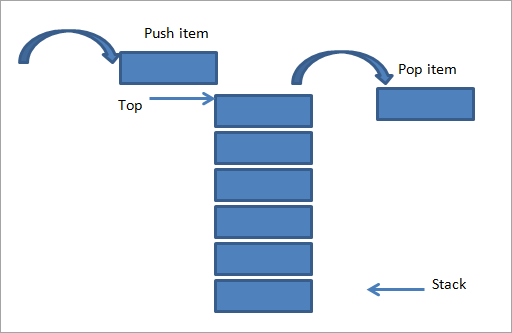
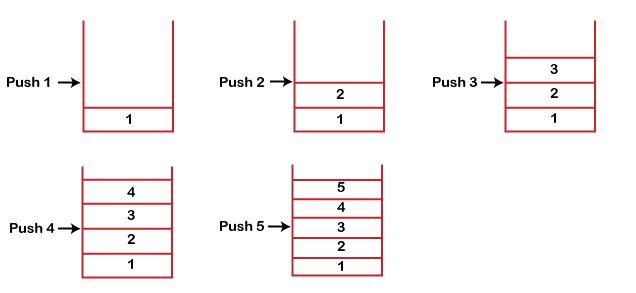
# What is a Stack?

A Stack is a linear data structure that follows the LIFO (Last-In-First-Out) principle. Stack has one end, whereas the Queue has two ends (front and rear). It contains only one pointer top pointer pointing to the topmost element of the stack. Whenever an element is added in the stack, it is added on the top of the stack, and the element can be deleted only from the stack. In other words, a stack can be defined as a container in which insertion and deletion can be done from the one end known as the top of the stack.





### Some key points related to stack

– It is called as stack because it behaves like a real-world stack, piles of books, etc.

– A Stack is an abstract data type with a pre-defined capacity, which means that it can store the elements of a limited size.

– It is a data structure that follows some order to insert and delete the elements, and that order can be LIFO or FILO.

### Standard Stack Operations

push(): When we insert an element in a stack then the operation is known as a push. If the stack is full then the overflow condition occurs.

* pop(): When we delete an element from the stack, the operation is known as a pop. If the stack is empty means that no element exists in the stack, this state is known as an underflow state.
* isEmpty(): It determines whether the stack is empty or not.
* isFull(): It determines whether the stack is full or not.'
* peek(): It returns the element at the given position.
* count(): It returns the total number of elements available in a stack.
* change(): It changes the element at the given position.
* display(): It prints all the elements available in the stack

### **Applications of Stack**

– String reversal

– UNDO/REDO

– Recursion:

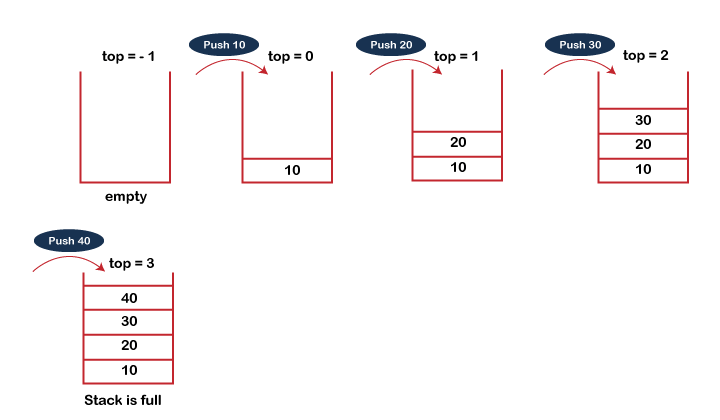
– DFS(Depth First Search)

– Backtracking:

– Expression conversion

– Memory management

### PUSH operation



### POP operation

