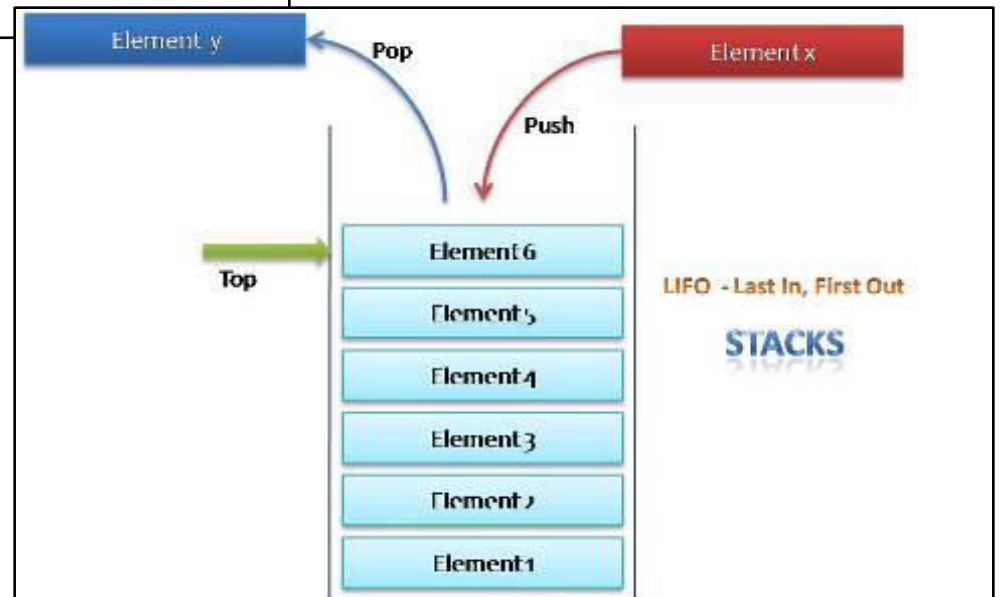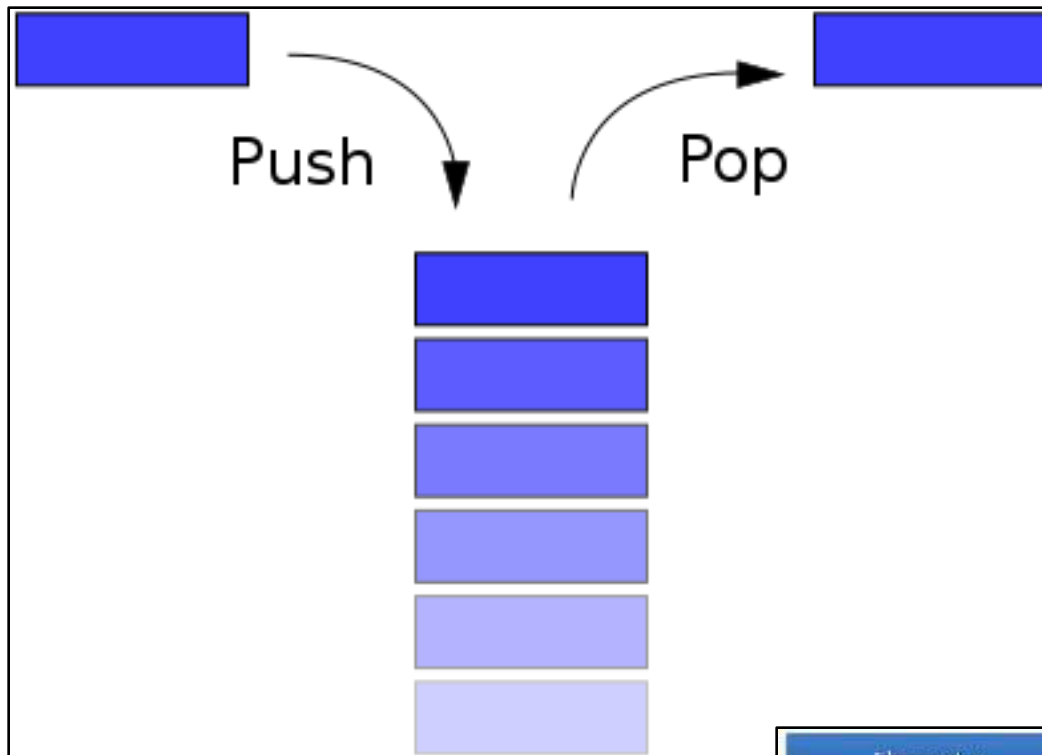# Chapter -3 [B]

stack

What is a stack?

❖ **A stack is a Last In, First Out (*LIFO*) data structure.**

❖ **Anything added to the stack goes on the "*top*" of the stack.**

❖ **Anything removed from the stack is taken from the "*top*" of the stack**

❖ **Things are removed in the reverse order from that in which they were inserted.**

Push

Pop

Element y

Pop

Element x

Push

Top → Element 6

Element 5

Element 4

Element 3

Element 2

Element 1

LIFO - Last In, First Out

STACKS

STACKS

- In the computers memory stack can be represented as a _linear array._

- Every stack has a variable TOP associated with it. TOP is used to store the address of the top most element of the stack. it is this position from where the element will be added or deleted.

- If _TOP = NULL_ then it indicates that stack is empty and if _TOP=MAX_ , then _stack is full._

- **STACK REPRESENTION:**

A	AB	ABC	ABCD	ABCDE			
0	1	2	3	TOP=4	5	6	7

STACKS

(1)push (this operation adds elements to the top of the stack)

(2)pop(this operation removes the element from top of the stack)

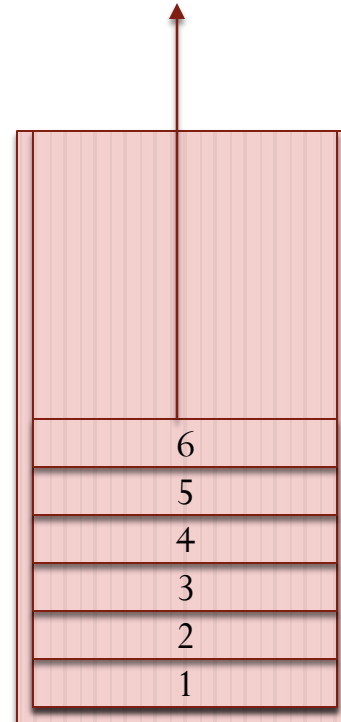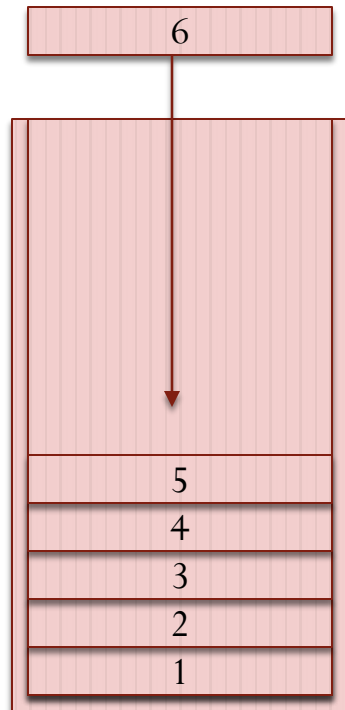(3)peep(returns the value (display)of topmost element of the stack)

Advantage:

last in, fist out(LIFO) access.

Disadvantage:

slow access to other elements.

Stack: Properties

- They both provide LIFO (last in first out) Structures.

Example of Stack (LIFO)

- A Stack of book on table.

- Token stack in Bank.

- Stack of trays and plates.