DAY-06:
- Operations SCLL, DLLL, DCLL
- Stack: concept & definition
- We can perform basic 3 operations on Stack in O(1) time: Push, Pop & Peek
- Stack can be implemented by 2 ways:
1. Static Stack (by using an array)
2. Dynamic Stack (by using linked list-dcll)

- there is no stack full condition in a dynamic stack
- if list is empty => stack is empty

Stack works in LIFO manner:
Push => addLast() - O(1)
Pop => deleteLast() - O(1)
Peek => get the data part of last node

if(head == null) => list is empty => stack is empty

head => 44 33 22 11

OR

Push => addFirst() - O(1)
Pop => deleteFirst() - O(1)
Peek => get the data part of first node

Lab Work => to implement dynamic stack

- stack is also used in / to implement expression conversion algorithms and evaluation algorithms.

What is an expression ?
Combination of operands and operators
- there are 3 types of expression
1. infix expression : a+b
2. prefix expression : +ab
3. postfix expression : ab+

infix expression => a*b/c*d+e-f*g+h

Lab Work : Implement algo to convert given parenthesized infix expression into its equivalent prefix expression.

Lab Work : Implement an algo for postfix evaluation (if an expression contains multi-digit operands).
- vector of strings to store infix & postfix expression
- stack of integers, while pushing an operand of type string we need to convert into its equivalent int.

+ Queue : it is a **linear/basic data structure** which is **collection/list of logically related similar type of data elements** in which elements can be added into it from one end referred as **rear end** and elements can be deleted from it which is at **front end.**

- in this collection/list, **element which was inserted first can be deleted first**, so this list works in **first in first out manner** or **last in last out manner**, hence queue is also called as **fifo list / lilo list.**
- On Queue data structure basic 2 operations can be performed in **O(1)** time:
1. Enqueue => insert/add an element into the queue from rear end.

2. Dequeue => delete/remove an element from the queue which is at front end.

- there are 4 types of queue:
1. linear queue (fifo)

2. circular queue (fifo)

3. priority queue => it is a type of queue in which elements can be added into it from rear end randomly (i.e. without checking priority), whereas element which is having highest priority can only be deleted first.

4. double ended queue (deque) => it is a type of queue in which elements can added as well as deleted from both the ends.
- on deque we can perform basic 4 operations in **O(1)** time:
i. push_back() => addLast()
ii. push_front() => addFirst()
iii. pop_back() => deleteLast()
iv. pop_front() => deleteFirst()
- deque can be implemented by using **DCLL.**
- further there are 2 types of deque:
1. input restricted deque => in this type of deque elements can be added into it only from one end, whereas elements can be deleted from both the ends.

2. output restricted deque => in this type of deque elements can be added into it from both the ends, whereas elements can be deleted only from one end.