Data Structures

- Data Structure is a way to store and organize data so that it can be used efficiently.
- There are two types of data structures:
  - ❖Primitive data structure
  - ❖Non-primitive data structure
- **Primitive Data structure**

The primitive data structures are primitive data types. The int, char, float, double, and pointer are the primitive data structures that can hold a single value.

- **Non-Primitive Data structure**

Derived from primitive data type

- The non-primitive data structure is divided into two types:
  - Linear data structure
  - Non-linear data structure
- **Linear Data Structure**
- The arrangement of data in a sequential manner is known as a linear data structure. The data structures used for this purpose are Arrays, Linked list, Stacks, and Queues. In these data structures, one element is connected to only one another element in a linear form.
- **Non-linear Data Structure**
- When one element is connected to the 'n' number of elements known as a non-linear data structure. The best example is trees and graphs. In this case, the elements are arranged in a random manner.

Major Operations

The major or the common operations that can be performed on the data structures are:

- **Searching:** We can search for any element in a data structure.
- **Sorting:** We can sort the elements of a data structure either in an ascending or descending order.
- **Insertion:** We can also insert the new element in a data structure.
- **Updation:** We can also update the element, i.e., we can replace the element with another element.
- **Deletion:** We can also perform the delete operation to remove the element from the data structure.

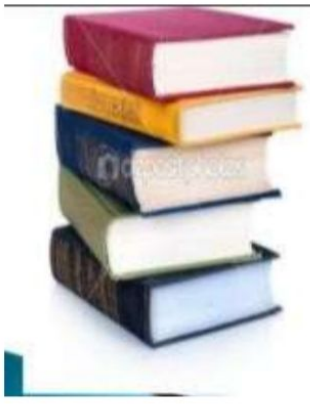
Array

- An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together.
- Represent as: Arr[0]

Stack

- Stack is a linear data structure which follows a particular order in which the operations are performed.
- The order may be LIFO(Last In First Out) or FILO(First In Last Out).
- There are two ways to implement a stack:
  - Using array
  - Using linked list

Example of Stack



Operations on Stack

- **push()** – Pushing an element on the stack.
- **pop()** – Removingan element from the stack.
- **peek()** – get the top data element of the stack, without removing it.
- **isFull()** – check if stack is full.
- **isEmpty()** – check if stack is empty.

Application

- Expression Handling
- Balancing of symbols
- Backtracking Procedure
- Redo-undo features at many places like editors, photoshop.
- Forward and backward feature in web browsers