Stack

Stack is a linear data structure that follows the **Last In, First Out (LIFO)** principle which can be a collection of certain data type in which we can only add and retrieve/remove a piece of data from the top of the structure, one end only, which is always last or topmost element of the structure. So in stack we can only able to perform operation such as insertion and deletion from one end which position is pointed by a pointer or variable called tos or TOP. There will be a pointer or variable always points to the last or topmost element in a stack.

The "top" is **where all adding (push) and removing (pop) happens**. Basically TOP focuses on:

- Last element or Top of the stack
- And index positioning of element at the top of stack. (index position tracking)

Core Operations

The primary operations performed on a stack are:

- **push**: Adds a new element to the top of the stack. If the stack is full (in array-based implementations), it results in a *stack overflow* condition.
- **pop**: Removes and returns the topmost element from the stack. If the stack is empty, it results in a *stack underflow* condition.
- **peek (or top)**: Returns the value of the top element without removing it.
- **isEmpty**: Checks if the stack has any elements.
- **isFull**: Checks if the stack has reached its maximum capacity (for fixed-size implementations).

Note: It is primarily implemented using either **arrays** or **linked list**.

Stack Implementation using an Array:

https://github.com/Sandesh775/DSA_BCA_Course/blob/main/Data%20Structures/Stack/Stack_Implementation_using_Array/Implementing_Stack.java