**Lab 5**

**Stacks**

**Introduction:**

A stack is an ordered collection of homogeneous data elements where the insertion and deletion operations occur at one end only, called the top of the stack. It has a LIFO (Last In First Out) behavior.

A **stack** is an Abstract **Data** Type (ADT), commonly used in most programming languages. It is named **stack** as it behaves like a real-world **stack**, for example – a deck of cards or a pile of plates, etc. A real-world **stack** allows operations at one end only.

A **stack** is a limited access data structure - elements can be added and removed from the **stack** only at the top. ... Push adds an item to the top of the **stack**, pop removes the item from the top.

**Objective:**

* The objective of the lab is to develop understanding of the Stack data structure and its basic functions.
* To use stacks to convert expressions from infix to postfix.

**APPLICATION:**

* **Stacks** can be used for expression evaluation.
* **Stacks** can be used to check parenthesis matching in an expression.
* **Stacks** can be used for Conversion from one form of expression to another.
* **Stacks** can be used for Memory Management.
* **Stack** data structures are used in backtracking problems.

**ISSUE:**

FACED ISSUES IN Q NO 1.

**CONCLUSION:**

Stacks is simple data structures that allow us to store and retrieve data sequentially. In a stack, the last item we enter is the first to come out. In a queue, the first item we enter is the first come out. We can add items to a stack using the push operation and retrieve items using the pop operation.