



## LAB 07

**Objective: Understanding the concept of Stacks and Queue using Array and LinkedList.**

Name: \_\_\_\_\_ Roll Number: \_\_\_\_\_  
Score: \_\_\_\_\_ Signature: \_\_\_\_\_ Date: 15/ 1/ 2024

LAB PERFOR MANCE INDICAT OR	SUBJECT KNOWLEDGE	DATA ANALYSIS AND INTERPRETAT ION	ABILITY TO CONDUCT EXPERIMENT	PR ES EN TA TION	CALCULA TION AND CODING	OBSE RVATI ON/RE SULTS	SCORE

### 1. Stacks

Stack is a container of objects that are inserted and removed according to the last-in first-out (LIFO) principle.

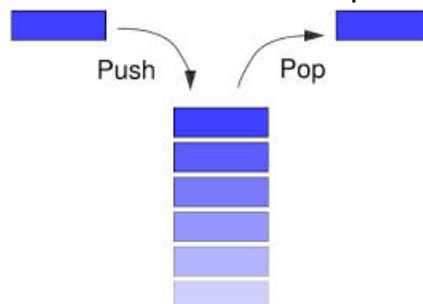
In the pushdown stacks only two operations are allowed:

- push** the item into the stack,
- pop** the item out of the stack.

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.

A helpful analogy is to think of a stack of books; you can remove only the top book, also you can add a new book on the top.



## Stack Operations:

**push():** Insert a new element into the stack

**pop():** Return the top element of the Stack

**peek():** Return the top element.

**display():** Print all elements in Stack.

## Implementation of Stack Using Array

### 1. Push operation

```
begin
  if top = n then stack full
    top = top + 1
    stack (top) := item;
end
```

### 2. Pop operation:

```
begin
  if top = 0 then stack empty;
    item := stack(top);
    top = top - 1;
end;
```

### 3. Peek operation:

```
Begin
  if top = -1 then stack empty
    item = stack[top]
  return item
End
```

## Lab Tasks

### 1. Implement Stack using Array

2. Implement Stack using Singly Linked List.