KATHMANDU UNIVERSITY

DHULIKHEL, KAVRE

School of Engineering

Lab Report No.: 02

Data Structures and Algorithm (COMP 202)

Submitted By:

Aayush Pokhrel

Roll No:42

Group: CE

Level: 2nd Year/1st Sem

Submitted To

Dr. Rajani Chulyadyo

Department of Computer Science And Engineering

Title: Stack Implementation

Objective: To Implement Stack from Array and Linked List

Functions To Use:

- push(element): Adds an element into the stack
- pop(): Removes an element from the stack
- isEmpty(): Checks if the stack is empty
- isFull(): Checks if the stack is full
- top(): Gives the element at the top

Something About Queue:

Stack" is a term commonly used in computer science and programming to refer to a data structure that follows the Last-In-First-Out (LIFO) principle. It is an abstract data type that represents a collection of elements, where elements can be inserted and removed only from one end called the "top" of the stack.

The operations performed on a stack typically include:

Push: Adds an element to the top of the stack.

Pop: Removes and returns the element at the top of the stack.

Peek or Top: Returns the element at the top of the stack without removing it.

IsEmpty: Checks whether the stack is empty or not.

Stacks are commonly used in various programming scenarios, such as function calls, expression evaluation, undo-redo operations, and backtracking algorithms. They provide an efficient way to manage and track the order of elements. empty or not.

- **Main.cpp Code**

```
#include "src/ArrayStack.cpp"
#include "src/StackLinkedList.cpp"
#include <iostream>
using namespace std;

int main()
{
    // Taking the size of the stack from the user
    int sizeofarray;
    cout << "Enter The size of stack" << endl;
    cin >> sizeofarray;

    // Creating the stack from array
    ArrayStack stack(sizeofarray);
    int value;
    for (int i = 0; i < sizeofarray; i++)
    {
        cout << "Enter The element to put inside stack" << endl;
        cin >> value;
        cout << "Adding " << value << " to the stack..." << endl;
        stack.push(value);
    }

    cout << "Adding 5 to the stack...But It Should Print Size limit Reached
since the size of array is " << sizeofarray << endl;
    cout << "Try Adding 5 to the stack..." << endl;
    stack.push(5);

    cout << "Displaying the topmost element of the Stack:" << endl;
    cout << stack.peek() << endl;

    cout << "Removing the topmost element of the Stack one by one:" << endl;
    for (int i = 0; i < sizeofarray; i++)
        cout << stack.pop() << endl;

    cout << "Since all the elemets are removed it cannot return any more" <<
endl;
    cout << stack.pop();

    // Stack from Linked List
    StackLinkedList stackLinked;
    cout << "" << endl;
    cout << "Implementing Stack from Linked List" << endl;
    cout << "" << endl;

    for (int i = 0; i < sizeofarray; i++)
    {
```

```
        cout << "Enter The element to put inside stack" << endl;
        cin >> value;
        stackLinked.push(value);
    }

    for (int i = 0; i < sizeofarray; i++)
    {
        cout << "Removing the topmost element of the Stack one by one:" <<
endl;
        cout << "peeking top most element: " << stackLinked.top() << endl;
        cout << "popped element: " << stackLinked.pop() << endl;
    }
}
```

Output

- **Array Stack Implementation**

```
Enter The size of stack
4
Enter The element to put inside stack
1
Adding 1 to the stack...
Enter The element to put inside stack
2
Adding 2 to the stack...
Enter The element to put inside stack
3
Adding 3 to the stack...
Enter The element to put inside stack
4
Adding 4 to the stack...
Adding 5 to the stack...But It Should Print Size limit Reached since the size of array is
Try Adding 5 to the stack...
Array Size Full
Displaying the topmost element of the Stack:
4
Removing the topmost element of the Stack one by one:
4
3
2
1
Since all the elemets are removed it cannot return any more
Stack is Empty
0
```

- **Linked Stack Implementation**

```
Implementing Stack from Linked List

Enter The element to put inside stack
4
Enter The element to put inside stack
5
Enter The element to put inside stack
6
Removing the topmost element of the Stack one by one:
peeking top most element: 6
popped element: 6
Removing the topmost element of the Stack one by one:
peeking top most element: 5
popped element: 5
Removing the topmost element of the Stack one by one:
peeking top most element: 4
popped element: 4
```

Github Link:[1014Aayush/LabWork (github.com)]