



THE UNIVERSITY  
OF LAHORE  
**ISLAMABAD  
CAMPUS**

## **Data Structure and Algorithms (CS13217)**

### **Lab Report**

Name: Muntaham Inaam Hashmi  
Registration #: SEU-F16-142  
Lab Report #: 03  
Dated: 30-04-2018  
Submitted To: Sir. Usman Ahmed

The University of Lahore, Islamabad Campus  
Department of Computer Science & Information Technology

## Experiment # 3

### Stack with Array implementation

#### Objective

The objective of this session is to understand the various operations in stack using array structure in C++.

#### Software Tool

1. Language: C++

## 1 Theory

Stacks are the most important in data structures. The notation of a stack in computer science is the same as the notion of the Stack to which you are accustomed in everyday life. For example, a recursion program on which function call itself, but what happen when a function which is calling itself call another function. Such as a function A call function B as a recursion. So, the firstly function B is call in A and then function A is work. So, this is a Stack. This is a Stack is First in Last Out data structure.

## 2 Task

### 2.1 Procedure: Task 3

Write a C++ code to perform : 1. Insertion in stack 2. Deletion in stack 3. Display the stack

Create a complete menu for the above options and also create option for reusing it.

```
#include<iostream>
using namespace std;
    int max=10;
    int a[10];
```

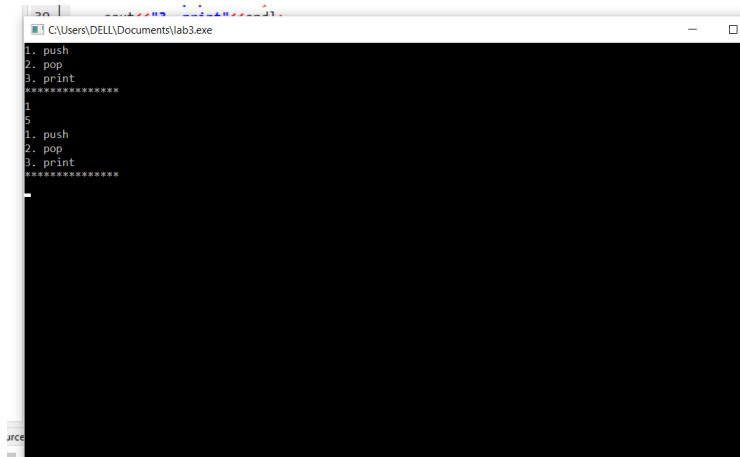


Figure 1: Link List

```

    int top=-1;
void push(int x)
{
    if (top==10)
    {
        cout<<" Error : Over _flow" ;
    }
    else
    {
        a[++top]=x;
    }
}
void pop()
{
    if (top== -1)
    {
        cout<<" Stack _is _empty" ;
    }
    else
    {
        top--;
    }
}
int print ()

```

```

        {
            for (int i=0;i<top+1;i++)
            {
                cout<<a[ i ];
            }
        }
    int main()
    {
        us: cout<<" 1. _push"<<endl;
        cout<<" 2. _pop"<<endl;
        cout<<" 3. _print"<<endl;
        cout<<" *****"<<endl;
        char k,y;
        while (y==0)
        {
            cin>>k;
            switch(k)
            {
                case '1':
                    int a;
                    cin>>a;
                    push(a);
                    goto us;
                    break;
                case '2':
                    pop();
                    goto us;
                    break;
                case '3':
                    print();
                    break;
            }
        }
    }
}

```

### **3 Conclusion**

In this lab we learned how to create stack and display it on a screen and its various functions.