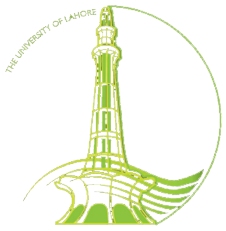

THE UNIVERSITY OF LAHORE
ISLAMABAD CAMPUS

Data Structure And Algorithum

Lab Report

Name:	Ch Hasnain Zafar
Registration #:	SEU-F16-104
Lab Report #:	03
Dated:	19-03-2018
Submitted To:	Mr. Usman Ahmed

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

Experiment # 1
Stack Implementation Using Array

Objective
To understand and implement the Stack Problem.

Software Tool
1. DEV C++

1 Theory

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO(First In Last Out).

Mainly the following three basic operations are performed in the stack:

Push: Adds an item in the stack. If the stack is full, then it is said to be an Overflow condition.
Pop: Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an Underflow condition.
Peek or Top: Returns top element of stack.
isEmpty: Returns true if stack is empty, else false.

2 Task

2.1 Procedure: Task 1

```
#include <iostream>
```

```
using namespace std;
        int A[50];
        int top=-1;
    //int c=0;

void push(int n)
{
        if(top<50)
        {
                cout<<"PUSH : "<<n<<endl;

                A[++top]=n;

        }
}
void pop()
{
        if(top>-1)
        {
                cout<<"POP : "<<A[top]<<endl;
                A[top]=0;
                top--;
        }
}
void display()
{
        //int k=0;


        for( int k=0; k<=top;k++)
        {
                cout<<" "<<A[k];


        }
}
int main() {
        int n=0;
        int b=0;
```

```
        int y=0;
        for(int o=0;o<50;o++)
        {

        cout<<"\n 1.PUSH \n 2.POP \n 3.display";
        cin>>y;
        switch(y)
        {

        case 1:
                cout<<"HOW MANY VALUES YOU WANT TO ENTER";
        cin>>b;

      for(int l=0;l<b;l++)
      {
                cout<<"ENTER NUMBER TO PUSH : ";

        cin>>n;

    push(n);
          }

                break;

        case 2:
                pop();
                break;
        case 3:
                display();
                break;
}
}
        return 0;
}
```