



THE UNIVERSITY
OF LAHORE
**ISLAMABAD
CAMPUS**

Data Structures and Algorithms (CS09203)

Lab Report

Name: Farhan Naseer
Registration #: SEU-F16-125
Lab Report #: 03
Dated: 16-04-2018
Submitted To: Sir. Usman Ahmed

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

Experiment # 03

Stack with Array implementation

Objective

The objective of this session is to understand the various operations on stack using arrays structure in C++. **Software Tool**

1. I use Code Blocks with GCC compiler.

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.

Insertions in Stack: In Stacks, we know the array work, sometimes we need to modify it or add some element in it. For that purpose, we use insertion scheme. By the use of this scheme we insert any element in Stacks using array. In Stack, we maintain only one node which is called TOP. And Push terminology is used as insertions.

Deletion in Stack: In the deletion process, the element of the Stack is deleted on the same node which is called TOP. In stacks, its just deleting the index of the TOP element which is added at last. In Stacks Pop terminology is used as deletion.

Display of Stack: In displaying section, the elements of Stacks are being display by using loops and variables as a reverse order. Such that, last element is display at on first and first element enters display at on last. Algorithm for top of stack varying method 1. Declare and initialize necessary variables, eg top = -1, MAXSIZE etc. 2. For push operation, if top = MAXSIZE - 1 print "stack overflow" else top = top + 1; Read item from user stack[top] = item 3. For next push operation, goto step 2. 4. For pop

operation, If top = -1 print "Stack underflow" Else item = stack[top] top = top - 1 Display item 5. For next pop operation, goto step 4. 6. Stop

2 Task

2.1 Procedure: Task 5

Write a C++ code to perform insertion and deletion in stack using arrays applying the algorithms

2.2

```
#include <iostream>

using namespace std;
    int A[50];
    int top=-1;
void push(int n)
{
    if (top<50)
    {
        cout<<"PUSH:_ "<<n<<endl;
        A[++top]=n;
    }
    else
        cout<<"error:stack_is_full"<<endl;
}
void pop()
{
    if (top>-1)
    {
        cout<<"POP:_ "<<A[top]<<endl;
        A[top]=0;
        top--;
    }
    else
        cout<<"error:stack_is_empty"<<endl;
}
```

```

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

int main() {
    int n=0;
    int b=0;
    int y=0;
    char ch;

do{
    cout<<"\n 1. Insertion_in_stack\n 2. Deletion_in_stack\n 3. Display\n";
    cout<<"enter_your_choice=";
    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;
    }
    cout<<"Do_you_want_to_continue_Y/N=";
    cin>>ch;
}
while((ch=='y')||(ch=='Y'));
    return 0;
}

```

3 Conclusion

In today lab we have studied about stack which last in first out (LIFO) in this we have created three functions one for pushing values in stack second for deleting it and third for displaying complete stack.