



**THE UNIVERSITY
OF LAHORE
ISLAMABAD
CAMPUS**

Data Structure and Algorithm (CS09203)

Lab Report

Name: Muhammad Umer
Registration #: CSU-F16-104
Lab Report #: 01
Dated: 26-03-2018
Submitted To: Mr. 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 on stack using arrays structure in C++.

Software Tool

1. Window 7 (32-bit)
2. Sublime Text Editor
3. Dev C++

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

Lab Task:-

1. Insertion in stack
2. Deletion in stack
3. Display the stack

Solution:-

```
#include<iostream>
#include<conio.h>
#define SIZE 101
using namespace std;
int stack[SIZE];
int top = -1;

void push(){
    if(top == SIZE-1){
        cout<<"\n\nError: _Stack_Overflow!";
        cout<<"\n\nPress _any_key_to_continue....";
```

```

        getch();
        return;
    }
    else
        top++;
    int item;
    cout<<"\n\nEnter value to insert: ";
    cin>>item;
    stack[top] = item;
    cout<<"\n\nValue inserted Successfully";
    cout<<"\n\nPress any key to continue ....";
    getch();
}

void pop(){
    int item;
    if(top == -1){
        cout<<"\n\nError: Stack Underflow!";
        cout<<"\n\nPress any key to continue ....";
        getch();
        return;
    }
    else{
        item = stack[top];
        top--;
        cout<<"\n\n"<<item<<" is removed from stack!";
    }
    cout<<"\n\nPress any key to continue ....";
    getch();
}

void display(){
    if(top == -1){
        cout<<"\n\nError: Stack is Empty!";
        cout<<"\n\nPress any key to continue ....";
        getch();
        return;
    }
    cout<<"\n\nItems in stack\n\n";
    for(int i=0; i<=top; i++){

```

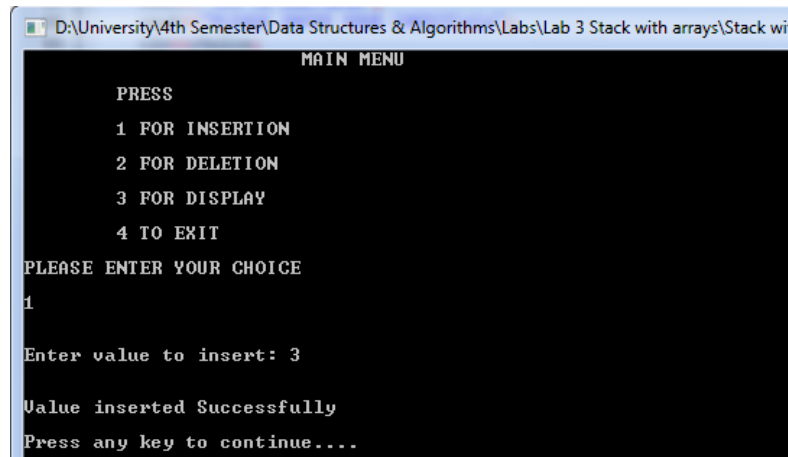
```

        cout<<stack[i]<<" ";
    }
    cout<<"\n\nPress any key to continue....";
    getch();
}

int main(){
    int choice;
up:
    system("cls");
    cout<<"\t\t\tMAIN_MENU\n\n";
    cout<<"\tPRESS\n\n";
    cout<<"\t1_FOR_INSERTION\n\n";
    cout<<"\t2_FOR_DELETION\n\n";
    cout<<"\t3_FOR_DISPLAY\n\n";
    cout<<"\t4_TO_EXIT\n\n";
    cout<<"PLEASE_ENTER_YOUR_CHOICE\n\n";
    cin>>choice;
    if(choice == 1){
        push();
        goto up;
    }
    else if(choice == 2){
        pop();
        goto up;
    }
    else if(choice == 3){
        display();
        goto up;
    }
    else if(choice == 4)
        exit(0);
    else{
        cout<<"\n\nWRONG_CHOICE!";
        cout<<"\n\nPRESS_ANY_KEY_TO_CHOOSE_AGAIN...";
        getch();
        goto up;
    }
    return 0;
}

```

Output:-



```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 3 Stack with arrays\Stack wit
MAIN MENU

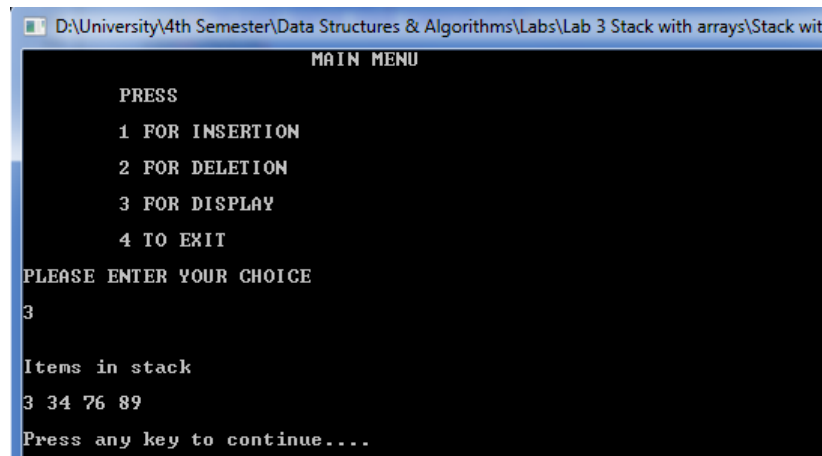
PRESS
1 FOR INSERTION
2 FOR DELETION
3 FOR DISPLAY
4 TO EXIT

PLEASE ENTER YOUR CHOICE
1

Enter value to insert: 3

Value inserted Successfully
Press any key to continue....
```

Figure 1: Main menu and insertion operation



```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 3 Stack with arrays\Stack wit
MAIN MENU

PRESS
1 FOR INSERTION
2 FOR DELETION
3 FOR DISPLAY
4 TO EXIT

PLEASE ENTER YOUR CHOICE
3

Items in stack
3 34 76 89

Press any key to continue....
```

Figure 2: Displaying after insertion

```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 3 Stack with arrays\Stack v

MAIN MENU

PRESS
1 FOR INSERTION
2 FOR DELETION
3 FOR DISPLAY
4 TO EXIT
PLEASE ENTER YOUR CHOICE
2
89 is removed from stack!
Press any key to continue....
```

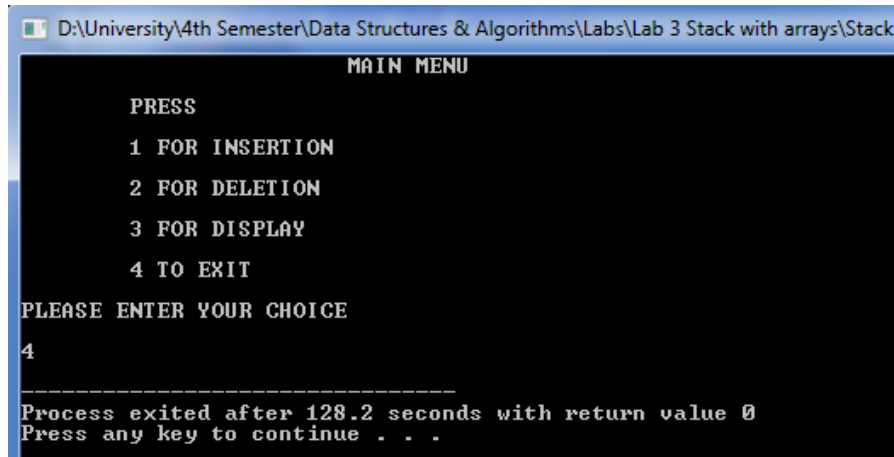
Figure 3: Deleting operation

```
D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 3 Stack with arrays\Stack with ...

MAIN MENU

PRESS
1 FOR INSERTION
2 FOR DELETION
3 FOR DISPLAY
4 TO EXIT
PLEASE ENTER YOUR CHOICE
3
Items in stack
3 34 76
Press any key to continue....
```

Figure 4: Displaying after deletion



```

D:\University\4th Semester\Data Structures & Algorithms\Labs\Lab 3 Stack with arrays\Stack
MAIN MENU

PRESS
1 FOR INSERTION
2 FOR DELETION
3 FOR DISPLAY
4 TO EXIT

PLEASE ENTER YOUR CHOICE
4

-----
Process exited after 128.2 seconds with return value 0
Press any key to continue . . .

```

Figure 5: Exit

Source Code:- <https://github.com/umerayan/Data-Structure-and-Algorithms>

Conclusion:- Stack in Data Structure work as First in Last out (FILO) concept. Stacks are the most important in data structures. Recursive function is one of the best example of stack. In this lab we implemented program for stack which is given above. The program is written in C++ and open source for everyone at my github account (The link provided above).

(Concerned Teacher/Lab Engineer)