18CS2007 Data Structures and Algorithms Lab

# Instructions:

1. Login on time to the lab session
2. The required softwares has to be installed in your system locally
3. Do not travel or sit in a place which disturbs the session
4. Any moment as per the request of the faculty member you have to show the screen and camera. Failing may leads to cancellation of attendance and lab mark
5. E-record and Videos are mandatory for all the exercises.

# Record:

* 1. You are expected to get your outputs verified in the respective lab.
  2. Strictly follow the given record template.
  3. Give your register number, exercise number, title and page number carefully.
  4. Output Screenshots must be taken with screen background color as white and foreground color as black. Font size must be 12. And font must be Times New Roman.
  5. Document alignment
     1. Page Size: A4
     2. Page Orientation: Portrait
     3. Margins: Normal with Bold border lines
     4. Font : Times New Roman, Normal
     5. Font Size: 12
     6. Line spacing is single line for program and 1.5 for Aim, Algorithm and Result.

|  |  |
| --- | --- |
| Ex.No:1A | Stack |
| Date:11/08/20 |

# Aim:

To make an array representation of stack in C++.

**Description:**

To perform push, pop and display operations.

**Algorithm:**

**1)PUSH**

**1.**Check whether stack is **full**.

**TOP=SIZE-1**

**2.**If stack is full, then display “**Stack is full”** and then terminate the loop.

**3.**If **not full**, increment the value of top by 1.

**TOP=TOP+1**

**4.**Assign a value to stack[top]

**STACK[TOP]=value**

**2)POP**

**1.**Check whether stack is empty

**TOP==-1**

**2.** If empty, display “**stack is empty, deletion is not possible”** and terminate.

**3.**If not empty, then define a variable and initialize it with top.

**4.**Display **stack[top].**

**5.**Decrement top by one.

**Top—**

**3)DISPLAY**

**1.**Check whether stack is empty

**TOP==-1**

**2.**If empty, display ”**stack is empty”** and terminate.

**3.**If not empty, initialize a variable and assign it to top and execute the loop until stack is equal to

Stop.

**4.**Display stack[i] and decrease the value of I by 1.

**Code:**

#include<iostream>

using namespace std;

//Array based stack

int stack[100],n=100,top=-1;

void push(int val){

if(top>n-1)

cout<<"Stack is Full!"<<"Insertion not possible"<<endl;

else{

top++;

stack[top]=val;

}

}

void pop(){

if(top<=-1)

cout<<"Stack is empty"<<endl;

else{

cout<<"The popped element is "<<stack[top]<<endl;

top--;

}

}

void display(){

if (top>=0){

cout<<"Stack elements are:";

for(int i=top;i>=0;i--)

cout<<stack[i]<<" ";

cout<<endl;

}else

cout<<"Stack is empty";

}

int main(){

int ch,val;

cout<<"1.Push in stack"<<endl;

cout<<"2.Pop from stack"<<endl;

cout<<"3.Display stack"<<endl;

cout<<"4.Exit"<<endl;

do{

cout<<"Enter choice :"<<endl;

cin>>ch;

switch(ch){

case 1:{

cout<<"Enter value to be pushed"<<endl;

cin>>val;

push(val);

break;

}

case 2:{

pop();

break;

}

case 3:{

display();

break;

}

case 4:{

cout<<"Exit"<<endl;

break;

}

default:{

cout<<"Invalid choice"<<endl;

}

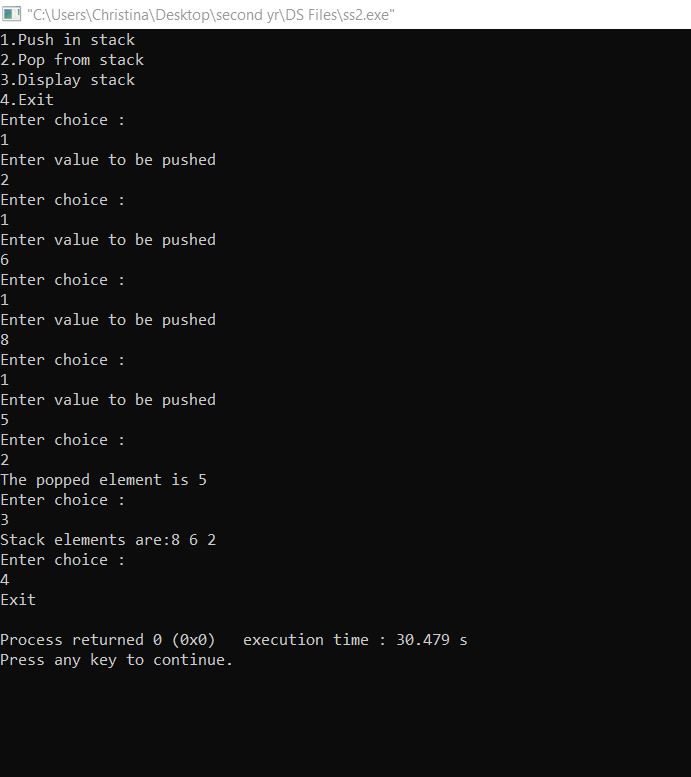
}

}while(ch!=4);

return 0;

}

**Output:**

**:**

**Video Link:** <https://www.youtube.com/watch?v=cp8XthSuwYU&t=303s>

**Result:** Thus the program to implement stack using array is executed and verified successfully.