Lab 5

Q1:

#include<iostream>

using namespace std;

const int STACKSIZE = 5;

class stack{

private:

int stackArray[STACKSIZE];

int TOP;

public:

stack(){

TOP = -1;

}

bool isEmpty(){

if(TOP == -1){

return true;

}else{

return false;

}

}

bool isFull(){

if (TOP == STACKSIZE - 1)

{

return true;

}else{

return false;

}

}

void Push(int element){

if (isFull() == true)

{

cout<<"Can't insert Element because the stack is already Full";

}else{

TOP++;

stackArray[TOP] = element;

}

}

void Pop(){

if (isEmpty() == true)

{

cout<<"Cannot remove the element Because the stack is already Empty";

}else{

TOP--;

}

}

int size(){

return TOP +1;

}

void DisplayStack(){

if (TOP == 0)

{

cout<<"No element in the Stack";

return;

}else{

for (int i = size()-1; i >=0 ; i--)

{

cout<<stackArray[i]<<endl;

}

}

}

};

int main(){

stack st;

st.Push(4);

st.Push(5);

st.Push(6);

st.DisplayStack();

st.Pop();

cout<<endl;

cout<<endl;

st.DisplayStack();

}

Q2:

#include<iostream>

using namespace std;

const int STACKSIZE = 5;

class stack{

private:

int stackArray[STACKSIZE];

int TOP;

public:

stack(){

TOP = -1;

}

bool isEmpty(){

if(TOP == -1){

return true;

}else{

return false;

}

}

bool isFull(){

if (TOP == STACKSIZE - 1)

{

return true;

}else{

return false;

}

}

void Push(int element){

if (isFull() == true)

{

cout<<"Can't insert Element because the stack is already Full";

}else{

TOP++;

stackArray[TOP] = element;

}

}

void Pop(){

if (isEmpty() == true)

{

cout<<"Cannot remove the element Because the stack is already Empty";

}else{

TOP--;

}

}

int size(){

return TOP +1;

}

void DisplayStack(){

if (TOP == 0)

{

cout<<"No element in the Stack";

return;

}else{

for (int i = size()-1; i >=0 ; i--)

{

cout<<stackArray[i]<<endl;

}

}

}

};

int main(){

stack st;

st.Push(100);

st.Push(200);

st.Push(300);

st.Push(400);

st.Push(500);

st.DisplayStack();

st.Pop();

cout<<"Poping "<<endl;

st.DisplayStack();

}

