**DSA PRACTICAL**

**NAME : Khushi Panwar**

**1 . WAP TO IMPLEMENT STACK USING ARRAYS**

#include<iostream>

#include<iomanip>

using namespace std;

class stacktype{

int arr[100];

int top;

public :

stacktype(){

top=(-1);

}

void push(int num);

int pop();

void display();

int isEmpty();

int isFull();

// int linearSearch();

};

void stacktype::push(int num){

int ele;

for (int i=0; i<num; i++){

cout<<"Enter the element : ";

cin>>ele;

arr[++top]=ele;

}

return;

}

int stacktype::pop(){

int ele=arr[top--];

return ele;

}

int stacktype::isEmpty(){

int flag;

if (top==-1) flag=1;

else flag=0;

return flag;

}

int stacktype::isFull(){

int flag;

if (top==sizeof(arr)-1) flag=1;

else flag=0;

return flag;

}

void stacktype::display(){

cout<<"-> DISPLAYING THE ARRAY : "<<endl<<endl;

for (int i=0; i<=top; i++){

cout<<setw(3)<<arr[i];

}

return;

}

void operationList(){

cout<<setw(20)<<"======================="<<endl;

cout<<"\t ARRAY OPERATIONS"<<endl;

cout<<"1. PUSH the element into stack"<<endl;

cout<<"2. POP the element from stack"<<endl;

cout<<"3. DISPLAY the elements of stack"<<endl;

cout<<"4. Check if stack is empty"<<endl;

cout<<"5. Check if stack is full"<<endl;

cout<<setw(20)<<"======================="<<endl<<endl;

return;

}

int main(){

cout<<setw(50)<<endl<<"\*\* STACK IMPLEMENTATION USING ARRAY \*\*"<<endl<<endl;

char ch='y';

int choice,ele, num,res;

stacktype obj;

while(ch=='y' || ch=='Y'){

operationList();

cout<<"--> Which operation do you want to perform :";

cin>>choice;

cout<<endl;

switch(choice){

case 1: cout<<"\t OPERATION CHOOSEN :: PUSH "<<endl<<endl;

res=obj.isFull();

if (res==1) cout<<"Error : Stack is full, Cannot push elements!"<<endl<<endl;

else {

cout<<"How many elements do you want to push: ";

cin>>num;

obj.push(num);

}

break;

case 2: cout<<"\t OPERATION CHOOSEN :: POP"<<endl<<endl;

res=obj.isEmpty();

if (res==1) cout<<"Error : Stack is empty, Cannot pop elements!"<<endl<<endl;

else {

ele=obj.pop();

cout<<setw(3)<<ele;

}

break;

case 3: cout<<"\t OPERATION CHOOSEN :: DISPLAY"<<endl<<endl;

obj.display();

break;

case 4: cout<<"\t OPERATION CHOOSEN :: isEmpty "<<endl<<endl;

res=obj.isEmpty();

if (res==1) cout<<"The stack is Empty"<<endl;

else cout<<"The stack is not Empty"<<endl;

break;

case 5: cout<<"\t OPERATION CHOOSEN :: isFull"<<endl<<endl;

res=obj.isFull();

if (res==1) cout<<"The stack is Empty"<<endl;

else cout<<"The stack is not Empty"<<endl;

break;

default : cout<<"\t Enter a valid choice!"<<endl;

}

cout<<"\n\n Do you want to continue? (y/n) :";

cin>>ch;

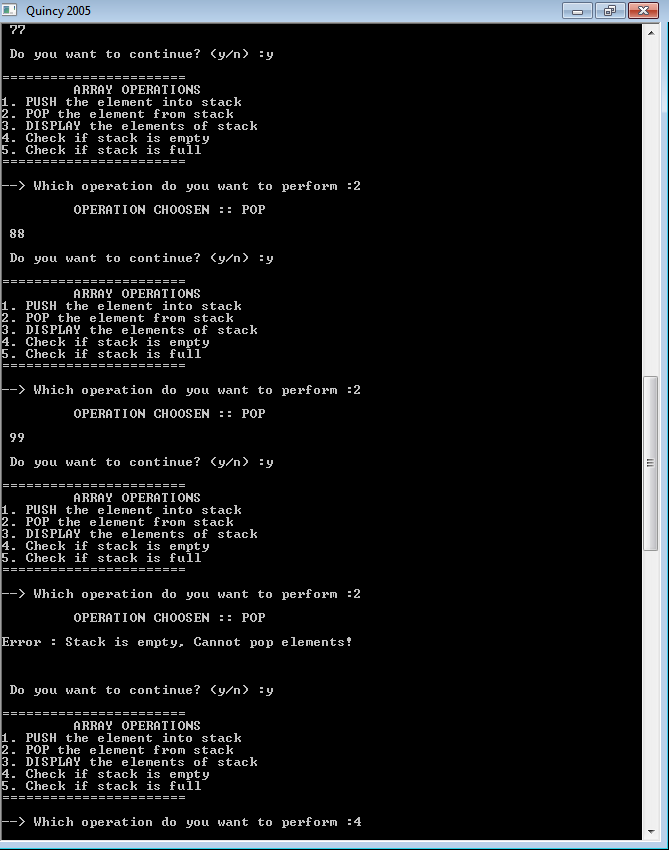
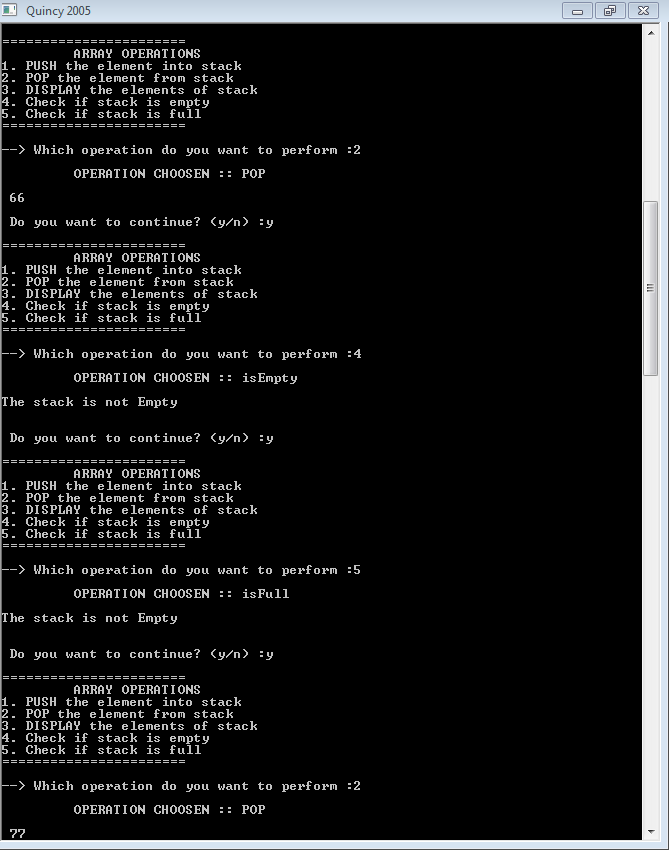
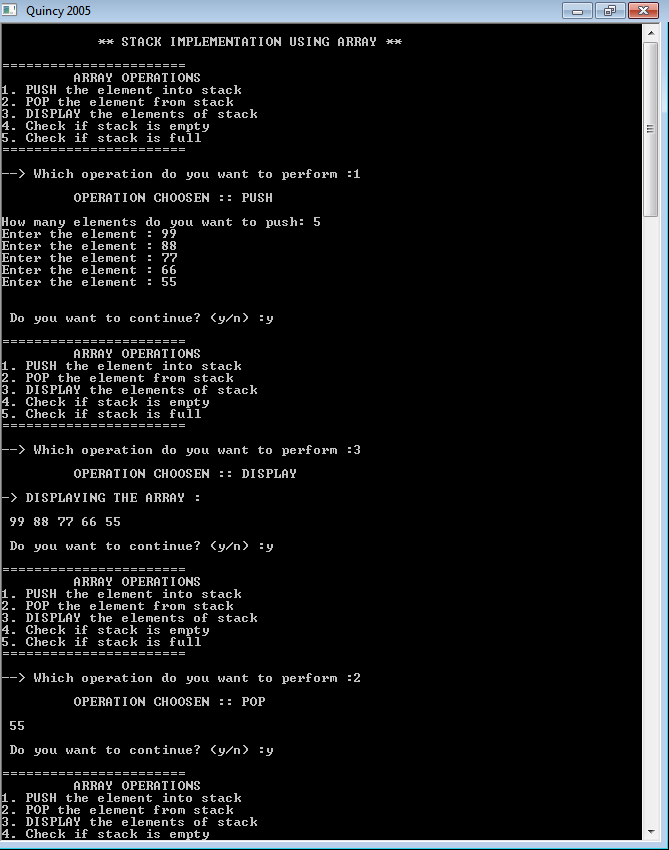
cout<<endl;

}

cout<<setw(20)<<"PROGRAM ENDS HERE"<<endl;

return 0;

}

s