

* ***Name*:**
* **Aqeel Abbas**
* ***Roll Number*:**
* **56**
* ***University*:**
* **Superior University**
* ***Submitted to***
* **: sir Rasikh sahab**

***Question no 1***

***.***Here is the **Lab 10 Question** along with the tasks:

### **Lab 10 – Stack with Linked List and Array**

#### **Objective:**

Understand and implement stack data structure using both array and linked list, and perform basic stack operations (Push, Pop, Display).

### **Tasks:**

**1. Stack using Array:**

* Implement a stack using a fixed-size array.
* Functions to implement:
  + push(int value) – insert element on top of stack
  + pop() – remove and return top element
  + display() – display all elements from top to bottom

**2. Stack using Linked List:**

* Implement a dynamic stack using singly linked list.
* Functions to implement:
  + push(int value) – insert element at head (top of stack)
  + pop() – remove and return head element
  + display() – display all nodes from top to bottom

Answer#include<iostream>

using namespace std;

#define SIZE 100

class StackArray {

int stack[SIZE];

int top;

public:

StackArray() {

top = -1;

}

void push(int value) {

if (top >= SIZE - 1) {

cout << "Stack Overflow!" << endl;

return;

}

stack[++top] = value;

}

void pop() {

if (top < 0) {

cout << "Stack Underflow!" << endl;

return;

}

cout << "Popped: " << stack[top--] << endl;

}

void display() {

if (top < 0) {

cout << "Stack is Empty!" << endl;

return;

}

cout << "Stack (top to bottom): ";

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

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

}

cout << endl;

}

};

int main() {

StackArray s;

s.push(10);

s.push(20);

s.push(30);

s.display();

s.pop();

s.display();

return 0;

}

Part 2#include<iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

};

class StackLinkedList {

Node\* top;

public:

StackLinkedList() {

top = NULL;

}

void push(int value) {

Node\* newNode = new Node();

newNode->data = value;

newNode->next = top;

top = newNode;

}

void pop() {

if (top == NULL) {

cout << "Stack Underflow!" << endl;

return;

}

cout << "Popped: " << top->data << endl;

Node\* temp = top;

top = top->next;

delete temp;

}

void display() {

if (top == NULL) {

cout << "Stack is Empty!" << endl;

return;

}

Node\* temp = top;

cout << "Stack (top to bottom): ";

while (temp != NULL) {

cout << temp->data << " ";

temp = temp->next;

}

cout << endl;

}

};

int main() {

StackLinkedList s;

s.push(5);

s.push(15);

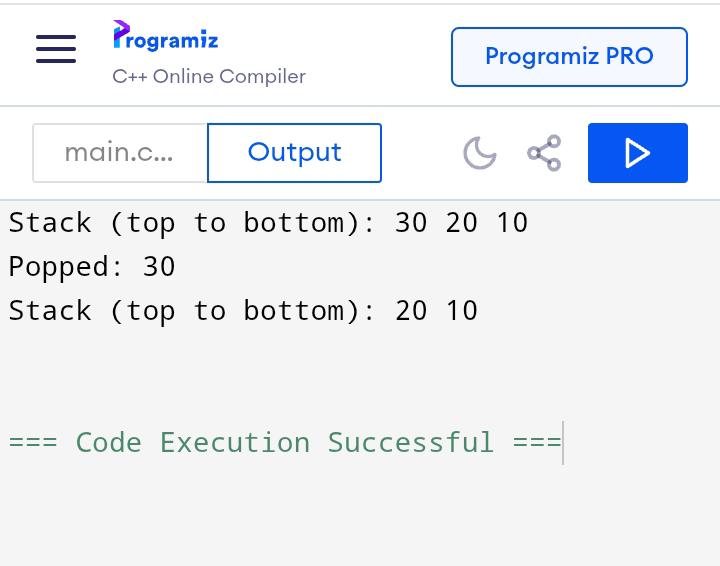
s.push(25);

s.display();

s.pop();

s.display();

return 0;

}