**EXPERIMENT-7**

**Aim:** Implement two stacks using a single array

**ALGORITHM:**

**SOURCE CODE:**

#include <stdio.h>

#include <liamits.h>

#include<iostream>

using namespace std;

#define LEFT\_STACK 0

#define RIGHT\_STACK 1

struct st {

int array[100];

int top1, top2;

} st;

void initialize() {

st.top1 = -1;

st.top2 = 100;

}

void push(int stack, int num) {

if(stack == LEFT\_STACK) {

if (st.top1 < st.top2-1) {

st.top1++;

st.array[st.top1] = num;

} else {

printf("Left Stack Full");

return;}

} else if(stack == RIGHT\_STACK) {

if (st.top1 < st.top2-1) {

st.top2--;

st.array[st.top2] = num;

} else {

printf("Right Stack Full");

return;}}

}

int pop(int stack) {

if(stack == LEFT\_STACK) {

if(st.top1 >= 0){

return st.array[st.top1--];

} else {

printf("Left Stack is Empty");

return INT\_MIN;}

}else if(stack == RIGHT\_STACK) {

if(st.top2 <= 99){

return st.array[st.top2++];

} else {

printf("Right Stack is Empty");

return INT\_MIN;}}

}

int main() {

initialize();

int a,b,c;

while (true){

cout<<"1.Left Stack 2.Right Stack 3.EXIT"<<endl;

cout<<"CHOOSE: ";cin>>a;

if (a==1){

cout<<"1.Push 2.Pop"<<endl;

cout<<"CHOOSE: ";cin>>b;

if (b==1){

cout<<"Enter Number:";cin>>c;

push(LEFT\_STACK,c);

}

else if(b==2){

printf("Pop from left stack %d\n", pop(LEFT\_STACK));}

}

else if(a==2){

cout<<"1.Push 2.Pop"<<endl;

cout<<"CHOOSE: ";cin>>b;

if (b==1){

cout<<"Enter Number:";cin>>c;

push(RIGHT\_STACK,c);}

else if(b==2){

printf("Pop from right stack %d\n", pop(RIGHT\_STACK));}

}

else {

exit(0);}}

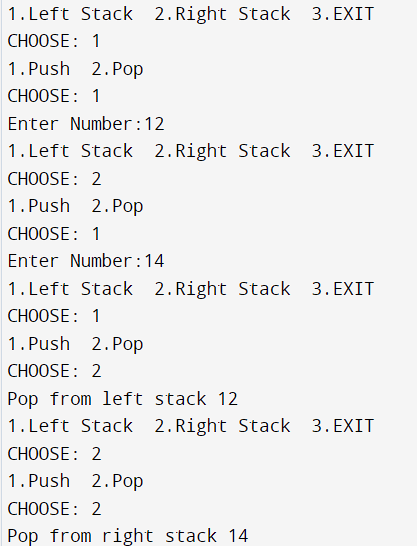
return 0;

}

**EXPERIMENT-7**

**Aim:** Implement two stacks using a single array.

**OUTPUT:**

****

**EXPERIMENT-8**

**Aim:** Create a stack and perform Push , Pop , and Display on stack using Linked List.

**ALGORITHM:**

**SOURCE CODE:**

#include <iostream>

using namespace std;

struct Node {

int data;

struct Node \*next;

};

struct Node\* top = NULL;

void push(int val) {

struct Node\* newnode = (struct Node\*) malloc(sizeof(struct Node));

newnode->data = val;

newnode->next = top;

top = newnode;

}

void pop() {

if(top==NULL)

cout<<"Stack Underflow"<<endl;

else {

cout<<"The popped element is "<< top->data <<endl;

top = top->next;

}

}

void display() {

struct Node\* ptr;

if(top==NULL)

cout<<"EMPTY STACK";

else {

ptr = top;

cout<<"Elements : ";

while (ptr != NULL) {

cout<< ptr->data <<" ";

ptr = ptr->next;

}

}

cout<<endl;

}

int main() {

int ch, val;

cout<<"1 Push"<<endl;

cout<<"2 Pop"<<endl;

cout<<"3 Display"<<endl;

cout<<"4 Exit"<<endl;

do {

cout<<"Enter choice: ";

cin>>ch;

switch(ch) {

case 1: {

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

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);

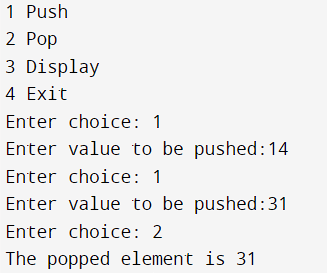
return 0;

}

**EXPERIMENT-8**

**Aim:** Create a stack and perform Push , Pop , and Display on stack using Linked List.

**OUTPUT:**

****