**DSA LAB – 8**

**Name:** Etcherla Sai Manoj **Mis. No:** 112015044 **Branch:** CSE

**Circular Queue:**

**Code:**

#include<iostream>

using namespace std;

#define SIZE 10

class CQueue{

int array[SIZE];

int rear;

int front;

public:

CQueue(){

rear = front = -1;

}

//function to check if queue is full

bool isFull(){

if(front == 0 && rear == SIZE - 1){

return true;

}

if(front == rear + 1) {

return true;

}

return false;

}

//function to check if queue is empty

bool isEmpty(){

if(front == -1) {

return true;

}

else {

return false;

}

}

//declaring insert\_q, delete\_q, display\_q and size functions

void insert\_q(int x);

int delete\_q();

void display\_q();

int size();

};

// function to insert element to queue

void CQueue :: insert\_q(int x){

if(isFull()){

cout << "Queue is full";

cout << "Queue OVERFLOW" << endl;

}

else{

if(front == -1){

front = 0;

}

rear = (rear + 1) % SIZE;

// inserting the element

array[rear] = x;

}

}

// function to delete element from queue

int CQueue :: delete\_q()

{

int y;

if(isEmpty()){

cout << "Queue is empty" << endl;

cout << "Queue UNDERFLOW" << endl;

return 0;

}

else{

y = array[front];

cout << "Deleted value from the queue is " << y << endl;

if(front == rear){

// only one element in queue, reset queue after removal

front = -1;

rear = -1;

}

else{

front = (front+1) % SIZE;

}

//cout << "Deleted value from the queue is " << array[front];

return(y);

}

}

// function to display elements of Circular Queue

void CQueue :: display\_q()

{

int i;

if(isEmpty()) {

cout << "Empty Queue" << endl;

}

else{

cout << "Front -> " << front;

cout << "\nElements -> ";

for(i = front; i != rear; i= (i+1) % SIZE){

cout << array[i] << "\t";

}

cout << array[i];

cout << "\nRear -> " << rear << endl;

}

}

int CQueue :: size()

{

if(rear >= front){

return (rear - front) + 1;

}

else{

return (SIZE - (front - rear) + 1);

}

}

int main()

{

CQueue q1;

int choice;

//Menu of queue operations

cout << "===================Menu===================\n";

cout << "1. Insert element to queue\n";

cout << "2. Delete element from queue\n";

cout << "3. Display elements of queue\n";

cout << "4. EXIT\n";

cout << "==========================================\n";

while(1){

cout << "\nEnter your choice to perform : ";

cin >> choice;

switch(choice)

{

case 1:

int v;

cout << "Enter value to insert : ";

cin >> v;

q1.insert\_q(v);

break;

case 2:

q1.delete\_q();

break;

case 3:

cout << "---Queue elements---\n";

q1.display\_q();

cout << "--------------------\n";

break;

case 4:

return 0;

default:

cout << "Enter valid choice...!!!\n";

break;

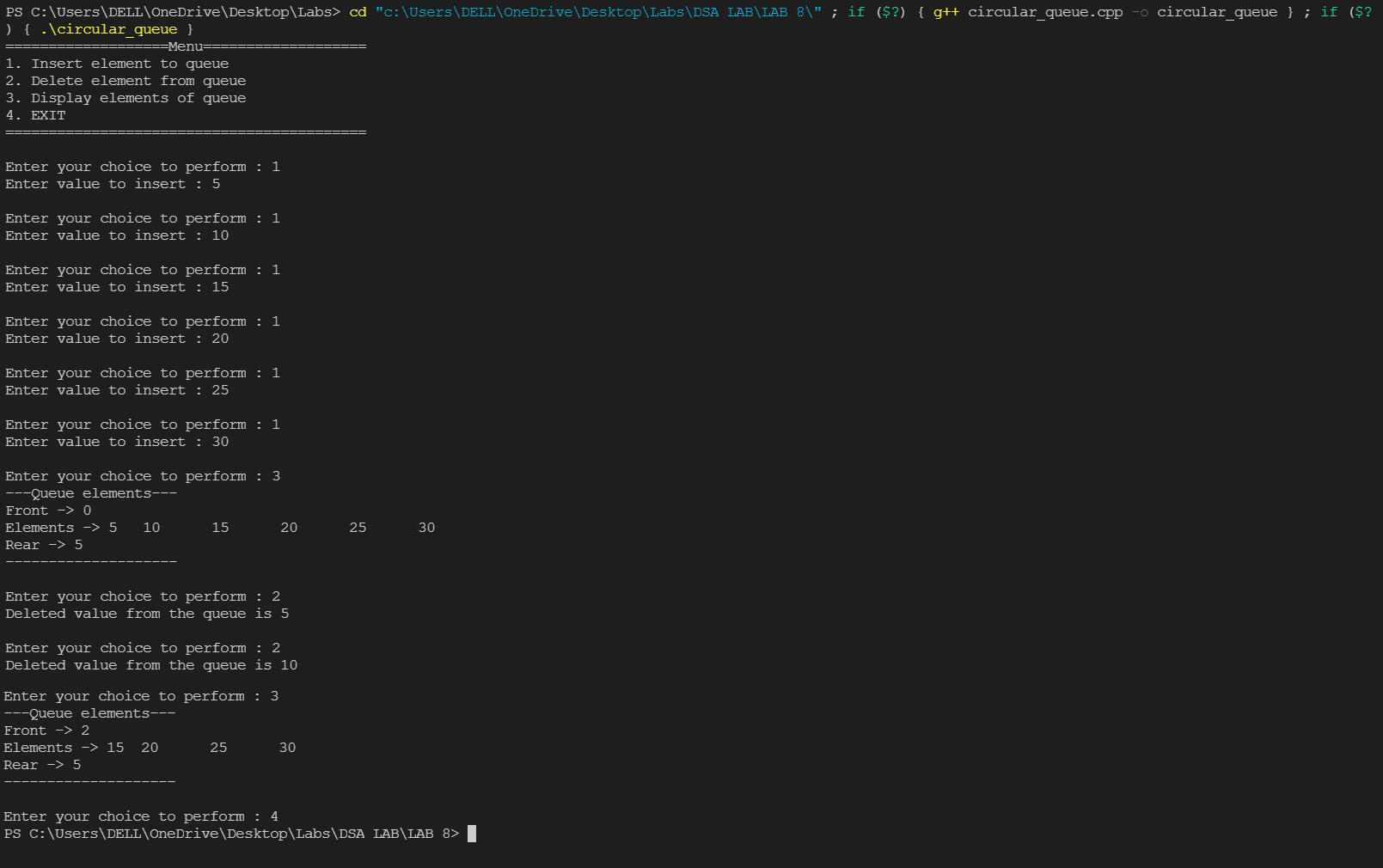
}

}

return 0;

}

**Input & Output:**

****

**Linear Queue:**

**Code:**

#include<iostream>

using namespace std;

#define SIZE 10

class Queue{

int array[SIZE];

int rear;

int front;

public:

Queue(){

rear = front = -1;

}

//declaring insert, delete, search and display functions

void insert\_q(int x);

int delete\_q();

void search\_q(int x);

void display\_q();

};

//function to insert element to queue

void Queue :: insert\_q(int x){

if(front == -1) {

front++;

}

if(rear == SIZE-1){

cout << "Queue OVERFLOW\n";

}

else{

array[++rear] = x;

}

}

//function to remove element from queue

int Queue :: delete\_q(){

if(front > rear){

cout << "Queue UNDERFLOW\n";

}

else{

cout << "Deleted value from queue is " << array[front] << endl;

}

return array[++front];

}

//function to search for array element in queue

void Queue :: search\_q(int x){

int flag = 0;

if(front == rear){

cout << "The Queue is empty\n";

}

else{

for(int i = front; i < SIZE; i++){

if(array[i] == x){

flag = 1;

break;

}

else{

flag = 0;

}

}

}

if(flag == 1){

cout << "THE ELEMENT IS PRESENT IN THE QUEUE\n";

}

else{

cout << "THE ELEMENT IS NOT PRESENT IN THE QUEUE\n";

}

}

//function to display queue elements

void Queue :: display\_q(){

for(int i = front; i <= rear; i++){

cout << array[i] << endl;

}

}

int main(){

Queue q1;

int choice;

//Menu of queue operations

cout << "===================Menu===================\n";

cout << "1. Insert element to queue\n";

cout << "2. Delete element from queue\n";

cout << "3. Search element in queue\n";

cout << "4. Display elements of queue\n";

cout << "5. EXIT\n";

cout << "==========================================\n";

while(1){

cout << "\nEnter your choice to perform : ";

cin >> choice;

switch(choice)

{

case 1:

int v;

cout << "Enter value to insert : ";

cin >> v;

q1.insert\_q(v);

break;

case 2:

q1.delete\_q();

break;

case 3:

int h;

cout << "Enter element to search in queue : ";

cin >> h;

q1.search\_q(h);

break;

case 4:

cout << "---Queue elements---\n";

q1.display\_q();

cout << "--------------------\n";

break;

case 5:

return 0;

default:

cout << "Enter valid choice...!!!\n";

break;

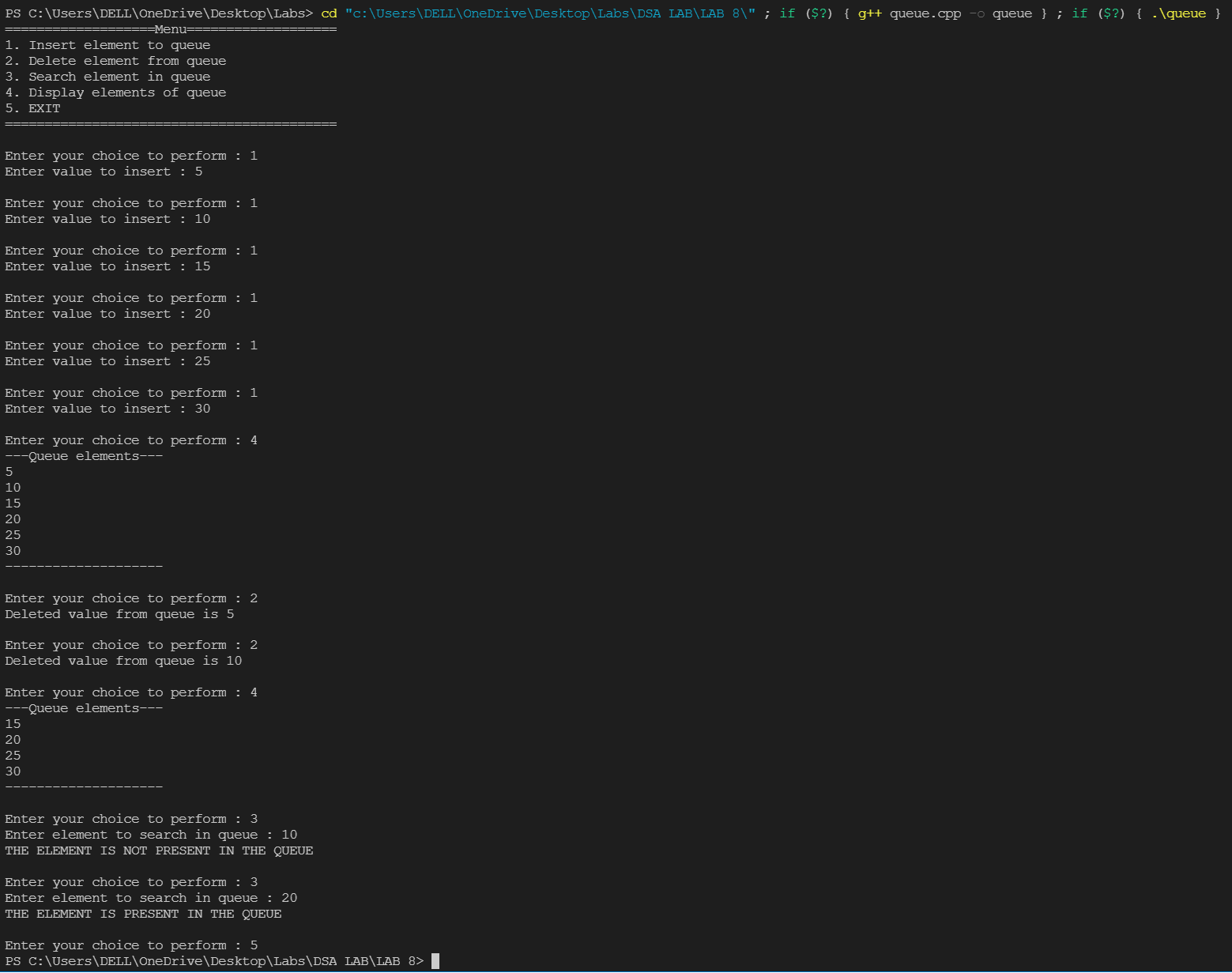
}

}

return 0;

}

**Input & Output:**

****