Practical – 11

AIM: Implement Queue using array Implement a Queue using an Array. Queries in the Queue are of the following type:

(i) 1 x (a query of this type means pushing 'x' into the queue)

(ii) 2 (a query of this type means to pop element from queue and print the poped element)

* Program

#include<bits/stdc++.h>

using namespace std;

int front=-1, rear=-1, Queue[5], len=5;

void enqueue(int);

int dequeue();

void display();

int main(){

int choice;

do{

cout << "1. Insert element in queue." << endl;

cout << "2. Delete element in queue." << endl;

cout << "3. Display" << endl;

cout << "4. exit" << endl;

cout << "Select the option : " ;

cin >> choice;

switch (choice)

{

case 1:

int x;

cout << "Enter a element you want to add : ";

cin >> x;

enqueue(x);

display();

break;

case 2:

int y;

y = dequeue();

cout << y << endl;

break;

case 3:

display();

break;

case 4:

break;

default:

break;

}

}while(choice!=4);

}

void enqueue(int x){

if(rear==len-1){

cout << "Queue is overflow" << endl;

return;

}

if(front==-1){

front=front+1;

rear = rear+1;

}

else{

rear++;

}

Queue[rear] = x;

}

int dequeue(){

int x;

if(front==-1){

cout << "Queue is underflow" << endl;

return 0;

}

else{

x = Queue[front];

if(front==rear){

front = -1;

rear = -1;

}

else

front++;

}

return x;

}

void display(){

cout << endl << "Elements : ";

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

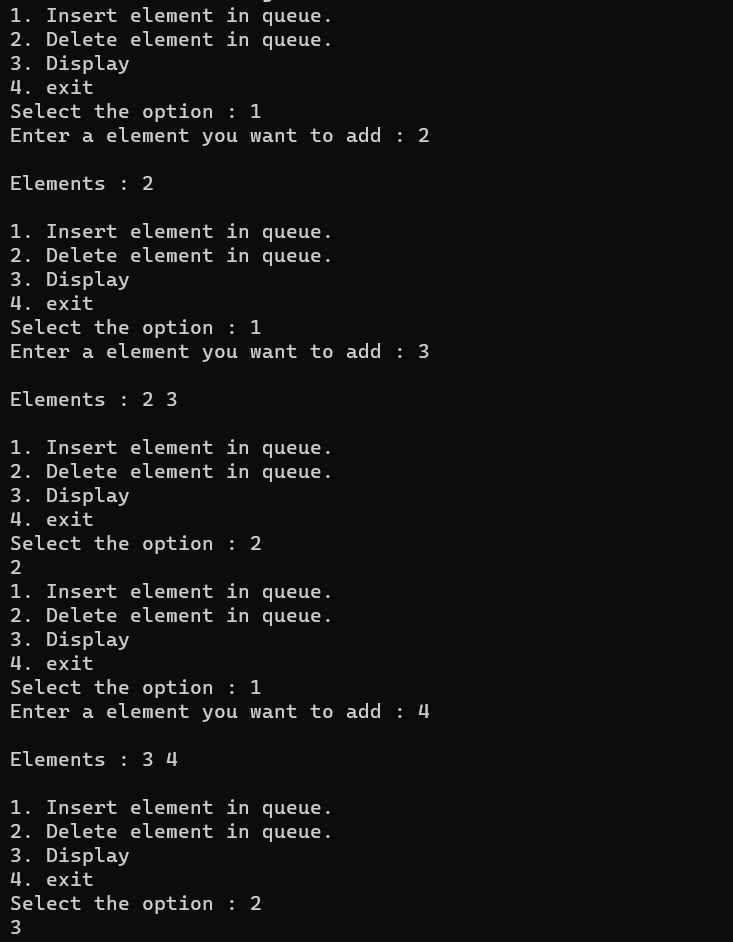
cout << Queue[i] << " ";

}

cout << endl << endl;

}

* Output



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

Student Signature Faculty Signature Marks