**Queues.h**

#pragma once

#include<iostream>

using namespace std;

class Queues

{

char\*arr;

int size;

public:

Queues();

Queues(char\*arrP,int s);

Queues(const Queues&obj);

~Queues();

void push(char data);

char pop();

bool isEmpty();

bool isFull();

int numOfElements();

void display();

char\*deepCopy(char\*p);

};

**Queues.cpp**

#include "Queues.h"

int front, rear;

char\*Queues::deepCopy(char\*p)

{

int length = strlen(p);

char\*temp = new char[length + 1];

for (int i = 0; i < length; i++)

{

temp[i] = p[i];

}

temp[length] = '\0';

return temp;

}

Queues::Queues()

{

arr = nullptr;

size = 0;

front = -1;

rear = -1;

}

Queues::Queues(char\*arrP, int s)

{

arr = deepCopy(arrP);

size = s;

front = -1;

rear = -1;

}

Queues::Queues(const Queues&obj)

{

arr = deepCopy(obj.arr);

size = obj.size;

}

Queues::~Queues()

{

}

bool Queues::isEmpty()

{

if (front == -1 && rear == -1)

{

return true;

}

else if (front == rear)

{

return true;

}

else

{

return false;

}

}

bool Queues::isFull()

{

if (rear == size - 1)

{

return true;

}

else

{

return false;

}

}

void Queues::push(char data)

{

if (isFull() == true)

{

cout << "Queue is Full." << endl;

}

rear++;

arr[rear] = data;

}

char Queues::pop()

{

front++;

if (isEmpty() == true)

{

cout << "Queue is Empty." << endl;

}

char data = arr[front];

return data;

}

int Queues::numOfElements()

{

int count = 0;

for (int i = front+1; i <= rear; i++)

{

count++;

}

return count;

}

void Queues::display()

{

for (int i = front+1; i <= rear; i++)

{

cout << arr[i] << " " << endl;

}

cout << endl;

}

**source.cpp**

#include"Queues.h"

int main()

{

const int siz = 5;

char ar[siz];

Queues obj(ar, siz);

obj.push('e');

obj.push('i');

obj.push('s');

obj.push('h');

obj.push('a');

obj.display();

cout << endl;

cout << "Deleted Elements are : ";

cout<<obj.pop();

cout << obj.pop();

cout << endl;

cout << endl;

cout << "After Deleted your Queue is : " << endl;

obj.display();

cout << endl;

int num = obj.numOfElements();

cout << "Number of Elements in Queue are : " << num << endl;

return 0;

}

**Output Console:**

****