

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

***Question no 1***

***.***Here is the **Lab 11 Question** with tasks:

## **Lab 11 – Queue with Array and Linked List**

### **Objective:**

To understand and implement the **Queue data structure** using both **Array** and **Linked List**, and perform standard queue operations.

### **Tasks:**

### **1. Queue using Array**

* Implement a queue using a fixed-size array.
* Functions to implement:
  + enqueue(int value) – add element at the rear
  + dequeue() – remove element from the front
  + display() – display all elements from front to rear

### **2. Queue using Linked List**

* Implement a dynamic queue using a singly linked list.
* Functions to implement:
  + enqueue(int value) – add element at the rear
  + dequeue() – remove element from the front
  + display() – display all elements from front to rear

Let me know if you want the Word file explanation with code and screenshots or help with Lab 12 next.

Answr#include<iostream>

using namespace std;

#define SIZE 100

class QueueArray {

int queue[SIZE];

int front, rear;

public:

QueueArray() {

front = rear = -1;

}

void enqueue(int value) {

if (rear == SIZE - 1) {

cout << "Queue Overflow!" << endl;

return;

}

if (front == -1) front = 0;

queue[++rear] = value;

}

void dequeue() {

if (front == -1 || front > rear) {

cout << "Queue Underflow!" << endl;

return;

}

cout << "Dequeued: " << queue[front++] << endl;

}

void display() {

if (front == -1 || front > rear) {

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

return;

}

cout << "Queue (front to rear): ";

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

cout << queue[i] << " ";

}

cout << endl;

}

};

int main() {

QueueArray q;

q.enqueue(10);

q.enqueue(20);

q.enqueue(30);

q.display();

q.dequeue();

q.display();

return 0;

}

Part 2. #include<iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

};

class QueueLinkedList {

Node\* front;

Node\* rear;

public:

QueueLinkedList() {

front = rear = NULL;

}

void enqueue(int value) {

Node\* newNode = new Node();

newNode->data = value;

newNode->next = NULL;

if (rear == NULL) {

front = rear = newNode;

} else {

rear->next = newNode;

rear = newNode;

}

}

void dequeue() {

if (front == NULL) {

cout << "Queue Underflow!" << endl;

return;

}

cout << "Dequeued: " << front->data << endl;

Node\* temp = front;

front = front->next;

if (front == NULL) rear = NULL;

delete temp;

}

void display() {

if (front == NULL) {

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

return;

}

cout << "Queue (front to rear): ";

Node\* temp = front;

while (temp != NULL) {

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

temp = temp->next;

}

cout << endl;

}

};

int main() {

QueueLinkedList q;

q.enqueue(5);

q.enqueue(15);

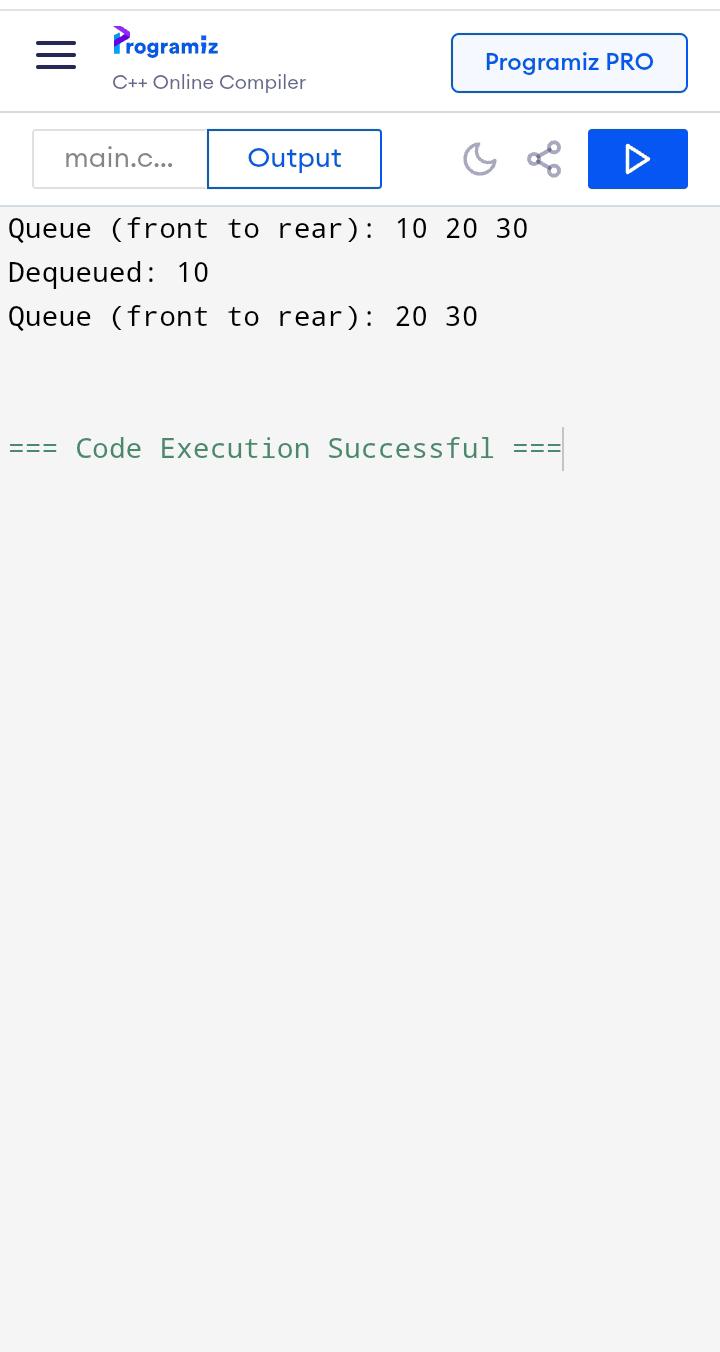
q.enqueue(25);

q.display();

q.dequeue();

q.display();

return 0;

}