**QUESTION NO: 11**

#include<iostream>

using namespace std;

class Node{

public:

int data;

Node\* next;

Node(int d = 0){

data = d;

next = nullptr;

}

};

class Queue{

Node\* front ,\*rear ;

public:

Queue(){

front = nullptr;

rear = nullptr;

}

void enqueue(int d){

Node\* newNode = new Node(d);

if(rear != nullptr){

rear->next = newNode;

rear = newNode;

}

else{

rear = front = newNode;

}

}

void dequeue(){

Node\* temp= front;

if(temp->next == nullptr){

temp = rear = nullptr;

}

front = front->next;

delete temp;

temp = nullptr;

}

void display(){

cout<<"Linked List: ";

Node\* temp = front;

while (temp != nullptr)

{

cout<<temp->data;

if(temp->next != nullptr){

cout<<" <> ";

}

temp = temp->next;

}

cout<<"\n";

}

};

class QueueWithArray{

int arr[100];

int count ;

public:

QueueWithArray(){

arr[0] = -1;

count = 0;

}

void enqueue(int d){

if(arr[0] == -1){

arr[0] = d;

count++;

}

else if(count == 99){

cout<<"Queue is Full.\n";

return;

}

else {

arr[count] = d;

count++;

}

}

void dequeue(){

arr[0] = 0;

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

{

arr[i] = arr[i+1];

}

count--;

}

void display(){

cout<<"Array: ";

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

{

cout<<arr[i];

if(i != count -1){

cout<<" <> ";

}

}

cout<<endl;

}

};

int main(){

cout<<"Queue Follow the First In first out(fifo) Rule \n\n";

cout<<"\tQueue With Linked List\n\n";

Queue q;

q.enqueue(99);

q.enqueue(89);

q.enqueue(79);

q.enqueue(819);

q.enqueue(59);

q.display();

cout<<"\nAfter Dequeue .\n";

q.dequeue();

q.display();

cout<<"\n\n\tQueue With Array: \n\n";

QueueWithArray qa;

qa.enqueue(22);

qa.enqueue(29);

qa.enqueue(99);

qa.enqueue(37);

qa.enqueue(19);

qa.enqueue(67);

qa.display();

cout<<"\nAfter Dequeue .\n";

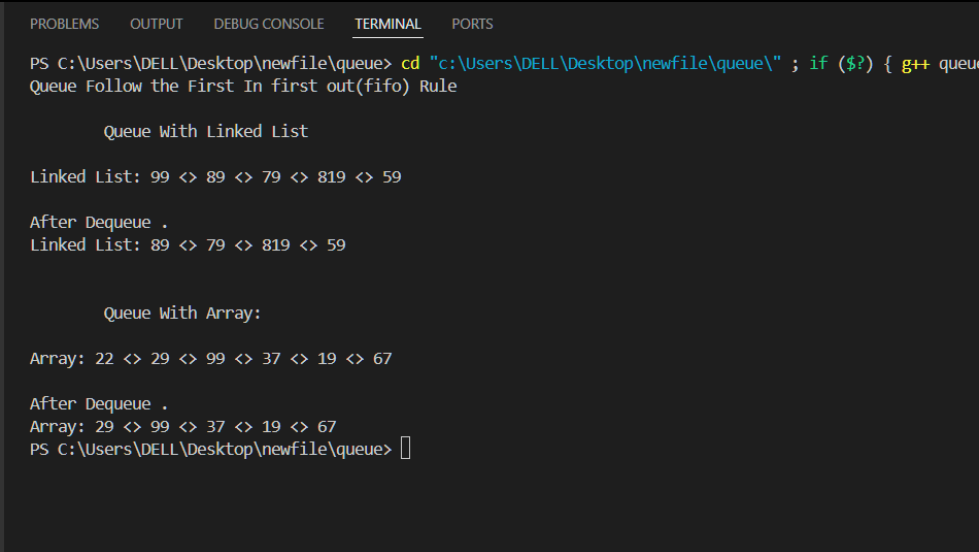
qa.dequeue();

qa.display();

return 0;

}

**OUTPUT**

****