

PRN: 2020BTEIT00041

Circular Queue:

```

1  /*
2  |   PRN: 2020BTEIT00041
3  |   Name: Om Vivek Gharge
4  | */
5
6  /*
7  |   Circular Queue - Implementation of a circular queue using an linked list
8  | */
9
10 #include <bits/stdc++.h>
11 using namespace std;
12
13 // Node class
14 class Node{
15 public:
16     int data;
17     Node *next;
18
19     Node(){
20         this->data = 0;
21         this->next = NULL;
22     }
23
24     Node(int data){
25         this->data = data;
26         this->next = NULL;
27     }
28 };
29
30 // Queue class
31 class Queue{
32 public:
33     Node* front;
34     Node* rear;
35
36     Queue(){
37         this->front = NULL;
38         this->rear = NULL;
39     }
40
41     void enqueue(int data);
42     int dequeue();
43     void Display();
44 };
45
46 // enqueue function
47 void Queue::enqueue(int data){
48     // create a new node

```

```

49     Node* new_node = new Node(data);
50
51     // if queue is empty
52     if(this->front == NULL){
53         // make hte new node as the front and rear
54         this->front = new_node;
55         this->rear = new_node;
56     }
57     // if queue is not empty
58     else{
59         // make the new node as the rear
60         this->rear->next = new_node;
61         this->rear = new_node;
62     }
63 }
64
65 // dequeue function
66 int Queue::dequeue(){
67     // if queue is empty
68     if(this->front == NULL){
69         cout << "Queue is empty" << endl;
70         return -1;
71     }
72     // if queue is not empty
73     else{
74         // store the front node in a temp variable
75         Node* temp = this->front;
76
77         // store the data of the front node
78         int data = temp->data;
79
80         // make the front as the next of the front
81         this->front = this->front->next;
82
83         // delete the temp node
84         delete temp;
85
86         // if the front is NULL
87         if(this->front == NULL){
88             // make the rear as NULL
89             this->rear = NULL;
90         }
91
92         return data;
93     }
94 }
95
96 void Queue::Display(){

```

```

97     // if queue is empty
98     if(this->front == NULL){
99         cout << "Queue is empty" << endl;
100        return;
101    }
102    // if queue is not empty
103    else{
104        // create a temp node
105        Node* temp = this->front;
106
107        // traverse the queue
108        while(temp != NULL){
109            cout << temp->data << " ";
110            temp = temp->next;
111        }
112        cout << endl;
113    }
114 }
115
116 int main(){
117     Queue q;
118
119     // Menu driven program to implement a circular queue
120     int choice;
121
122     do{
123         cout<<"-----Menu-----\n";
124         cout<<"1. Enqueue"<<endl;
125         cout<<"2. Dequeue"<<endl;
126         cout<<"3. Display"<<endl;
127         cout<<"4. Exit"<<endl;
128
129
130         cout<<"Enter your choice: ";
131         cin>>choice;
132
133         switch (choice){
134             case 1:
135                 int data;
136                 cout<<"Enter the data to be enqueued: ";
137                 cin>>data;
138                 q.enqueue(data);
139                 break;
140             case 2:
141                 cout<<"Dequeued element: "<<q.dequeue()<<endl;
142                 break;
143             case 3:
144                 cout<<"Display"<<endl;
145                 q.Display();
146                 break;
147             case 4:
148                 cout<<"Exiting..."<<endl;
149                 break;
150             default:
151                 cout<<"Invalid choice"<<endl;
152         }
153     }while(choice != 4);
154
155     return 0;
156 }

```

OUTPUT:

```
-----Menu-----
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 1
Enter the data to be enqueued: 1
-----Menu-----
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 1
Enter the data to be enqueued: 2
-----Menu-----
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 1
Enter the data to be enqueued: 3
-----Menu-----
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 3
Display
1 2 3
-----Menu-----
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 2
Dequeued element: 1
-----Menu-----
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 3
Display
2 3
-----Menu-----
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 4
Exiting...
```