

Run

Debug

Stop

Share

Save

Beautify

Language C

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define MAX_SIZE 5
4 typedef struct {
5     int items[MAX_SIZE];
6     int front;
7     int rear;
8 } CircularQueue;
9 void initializeQueue(CircularQueue *queue) {
10     queue->front = -1;
11     queue->rear = -1;
12 }
13 int isFull(CircularQueue *queue) {
14     return (queue->front == 0 && queue->rear == MAX_SIZE - 1) || (queue->rear == (queue->front - 1) % (MAX_SIZE - 1));
15 }
16 int isEmpty(CircularQueue *queue) {
17     return queue->front == -1;
18 }
19 void enqueue(CircularQueue *queue, int data) {
20     if (isFull(queue)) {
21         printf("Queue is full. Cannot enqueue.\n");
22     } else {
23         if (queue->front == -1) {
24             queue->front = 0;
25         }
26         queue->rear = (queue->rear + 1) % MAX_SIZE;
27         queue->items[queue->rear] = data;
28         printf("%d enqueued to the queue.\n", data);
29     }
30 }
31 int dequeue(CircularQueue *queue) {
32     int data;
33     if (isEmpty(queue)) {
34         printf("Queue is empty. Cannot dequeue.\n");
35         return -1;
36     } else {
37         data = queue->items[queue->front];
38         if (queue->front == queue->rear) {
39             queue->front = -1;
40             queue->rear = -1;
41         } else {
42             queue->front = (queue->front + 1) % MAX_SIZE;
43         }
44         return data;
45     }
46 }
```

Activate Windows  
Go to Settings to activate Windows.

```

34     printf("Queue is Empty. Cannot dequeue.\n");
35     return -1;
36 } else {
37     data = queue->items[queue->front];
38     if (queue->front == queue->rear) {
39         queue->front = -1;
40         queue->rear = -1;
41     } else {
42         queue->front = (queue->front + 1) % MAX_SIZE;
43     }
44     return data;
45 }
46 }
47 void displayQueue(CircularQueue *queue) {
48     int i;
49     if (isEmpty(queue)) {
50         printf("Queue is empty.\n");
51     } else {
52         printf("Front: %d, Rear: %d\n", queue->front, queue->rear);
53         printf("Queue elements: ");
54         for (i = queue->front; i != queue->rear; i = (i + 1) % MAX_SIZE) {
55             printf("%d ", queue->items[i]);
56         }
57         printf("%d\n", queue->items[i]);
58     }
59 }
60 int main() {
61     CircularQueue queue;
62     initializeQueue(&queue);
63     enqueue(&queue, 10);
64     enqueue(&queue, 20);
65     enqueue(&queue, 30);
66     enqueue(&queue, 40);
67     enqueue(&queue, 50);
68     enqueue(&queue, 60);
69     displayQueue(&queue);
70     int item = dequeue(&queue);
71     if (item != -1) {
72         printf("Dequeued item: %d\n", item);
73     }
74     displayQueue(&queue);
75     enqueue(&queue, 70);
76     displayQueue(&queue);
77     return 0;
78 }
79

```

input

```
10 enqueued to the queue.  
20 enqueued to the queue.  
30 enqueued to the queue.  
40 enqueued to the queue.  
50 enqueued to the queue.  
Queue is full. Cannot enqueue.  
Front: 0, Rear: 4  
Queue elements: 10 20 30 40 50  
Dequeued item: 10  
Front: 1, Rear: 4  
Queue elements: 20 30 40 50  
70 enqueued to the queue.  
Front: 1, Rear: 0  
Queue elements: 20 30 40 50 70
```

```
...Program finished with exit code 0  
Press ENTER to exit console.
```

Activate Windows  
Go to Settings to activate Windows.