

Experiment 03\circular_queue.c

```
1 #include <stdio.h>
2 #define MAX 5
3
4 int queue[MAX];
5 int front = -1, rear = -1;
6
7 void enqueue(int value) {
8     if ((front == 0 && rear == MAX - 1) || (rear == (front - 1) % (MAX - 1))) {
9         printf("Queue Overflow\n");
10        return;
11    }
12    if (front == -1) {
13        front = rear = 0;
14        queue[rear] = value;
15    } else if (rear == MAX - 1 && front != 0) {
16        rear = 0;
17        queue[rear] = value;
18    } else {
19        rear++;
20        queue[rear] = value;
21    }
22    printf("%d enqueue\n", value);
23 }
24
25 void dequeue() {
26     if (front == -1) {
27         printf("Queue Underflow\n");
28         return;
29     }
30     printf("%d dequeued\n", queue[front]);
31     if (front == rear) {
32         front = rear = -1;
33     } else if (front == MAX - 1) {
34         front = 0;
35     } else {
36         front++;
37     }
38 }
39
40 void display() {
41     if (front == -1) {
42         printf("Queue is empty\n");
43         return;
44     }
45     printf("Queue: ");
46     if (rear >= front) {
47         for (int i = front; i <= rear; i++)
48             printf("%d ", queue[i]);
```

```
49 } else {
50     for (int i = front; i < MAX; i++)
51         printf("%d ", queue[i]);
52     for (int i = 0; i <= rear; i++)
53         printf("%d ", queue[i]);
54 }
55 printf("\n");
56 }
57
58 int main() {
59     int choice, value;
60     while (1) {
61         printf("\n1. Enqueue 2. Dequeue 3. Display 4. Exit\n");
62         printf("Enter choice: ");
63         scanf("%d", &choice);
64         switch (choice) {
65             case 1:
66                 printf("Enter value: ");
67                 scanf("%d", &value);
68                 enqueue(value);
69                 break;
70             case 2:
71                 dequeue();
72                 break;
73             case 3:
74                 display();
75                 break;
76             case 4:
77                 return 0;
78             default:
79                 printf("Invalid choice\n");
80         }
81     }
82 }
83 }
```