

```
1 #include <stdio.h>
2
3 #define MAX_SIZE 5
4
5 int queue[MAX_SIZE];
6 int front = -1, rear = -1;
7
8 // Function to check if the queue is empty
9 int isEmpty() {
10     return front == -1 && rear == -1;
11 }
12
13 // Function to check if the queue is full
14 int isFull() {
15     return (rear + 1) % MAX_SIZE == front;
16 }
17
18 // Function to insert an element into the circular queue
19 void insert(int item) {
20     if (isFull()) {
21         printf("Queue Overflow: Cannot insert element %d\n", item);
22         return;
23     }
24
25     if (isEmpty()) {
26         front = rear = 0;
27     } else {
28         rear = (rear + 1) % MAX_SIZE;
29     }
30
31     queue[rear] = item;
32     printf("Element %d inserted successfully\n", item);
33 }
34
35 // Function to delete an element from the circular queue
36 void delete() {
37     if (isEmpty()) {
38         printf("Queue Underflow: Cannot delete from an empty queue\n");
```

```

    return;
}

printf("Element %d deleted successfully\n", queue[front]);

if (front == rear) {
    front = rear = -1;
} else {
    front = (front + 1) % MAX_SIZE;
}
}

// Function to display the elements of the circular queue
void display() {
    if (isEmpty()) {
        printf("Queue is empty\n");
        return;
    }

    printf("Elements in the queue: ");
    int i = front;
    do {
        printf("%d ", queue[i]);
        i = (i + 1) % MAX_SIZE;
    } while (i != (rear + 1) % MAX_SIZE);

    printf("\n");
}

```

```
int main() {  
    insert(1);  
    insert(2);  
    insert(3);  
    insert(4);  
    insert(5);  
  
    // Uncomment the line below to test queue overflow condition  
    // insert(6);  
  
    display();  
  
    delete();  
    delete();  
    delete();  
  
    // Uncomment the line below to test queue underflow condition  
    // delete();  
  
    display();  
  
    insert(6);  
    insert(7);  
  
    display();  
  
    return 0;  
}
```