```cpp
/*Implement circular queue using arrays. */
#include <iostream>
using namespace std;

class CircularQueue {
    int *queue, size, front, rear;

public:
    CircularQueue(int s) {
        size = s;
        queue = new int[size];
        front = rear = -1;
    }
    void enqueue(int x);
    int dequeue();
    void display();
};

void CircularQueue::enqueue(int x) {
    if ((front == 0 && rear == size - 1) || (front == rear + 1)) {
        cout << "Queue is full\n";
        return;
    }
    else if (front == -1) {
        front = rear = 0;
    }
    else if (rear == size - 1 && front != 0) {
        rear = 0;
    }
    else {
        rear++;
    }
    queue[rear] = x;
}

int CircularQueue::dequeue() {
    if (front == -1) {
        cout << "Queue is empty\n";
        return -1;
    }

    int x = queue[front];
    if (front == rear) {
        front = rear = -1;
    }
    else if (front == size - 1) {
        front = 0;
    }
    else {
        front++;
    }
    return x;
}

void CircularQueue::display() {
    if (front == -1) {
        cout << "Queue is empty\n";
        return;
    }
    if (rear >= front) {
        for (int i = front; i <= rear; i++)
            cout << queue[i] << " ";
    }
    else {
        for (int i = front; i < size; i++)
            cout << queue[i] << " ";
        for (int i = 0; i <= rear; i++)
            cout << queue[i] << " ";
    }
}

int main() {
    CircularQueue q(5);
    q.enqueue(1);
    q.enqueue(2);
    q.enqueue(3);
    q.enqueue(4);
    q.enqueue(5);
    q.enqueue(6);
    q.display();
    cout << endl;
    q.dequeue();
    q.dequeue();
    q.display();
    cout << endl;
    return 0;
}
```