```
#include <iostream>
using namespace std;

class CircularQueue {
private:
    int* queue;
    int front, rear, size, capacity;

public:
    CircularQueue(int cap) : capacity(cap), size(0), front(-1), rear(-1) {
        queue = new int[cap];
    }

    ~CircularQueue() {
        delete[] queue;
    }

    bool isFull() {
        return size == capacity;
    }

    bool isEmpty() {
        return size == 0;
    }

    void placeOrder(int orderID) {
        if (isFull()) {
            cout << "Queue is full, cannot place order." << endl;
            return;
        }
        if (front == -1) {
            front = 0;
```

```
        }
        rear = (rear + 1) % capacity;
        queue[rear] = orderID;
        size++;
        cout << "Order placed: " << orderID << endl;
    }

    void serveOrder() {
        if (isEmpty()) {
            cout << "Queue is empty, no orders to serve." << endl;
            return;
        }
        cout << "Order served: " << queue[front] << endl;
        front = (front + 1) % capacity;
        size--;
    }

    void displayOrders() {
        if (isEmpty()) {
            cout << "No orders in the queue." << endl;
            return;
        }
        cout << "Current orders in the queue: ";
        for (int i = 0; i < size; i++) {
            cout << queue[(front + i) % capacity] << " ";
        }
        cout << endl;
    }
};

int main() {
    int capacity;
```

```
cout << "Enter the maximum number of orders the pizza parlor can accept: ";
cin >> capacity;

CircularQueue cq(capacity);
int choice, orderID;

while (true) {
    cout << "\nPizza Parlor Menu:" << endl;
    cout << "1. Place Order" << endl;
    cout << "2. Serve Order" << endl;
    cout << "3. Display Orders" << endl;
    cout << "4. Exit" << endl;
    cout << "Enter your choice: ";
    cin >> choice;

    switch (choice) {
        case 1:
            cout << "Enter Order ID: ";
            cin >> orderID;
            cq.placeOrder(orderID);
            break;
        case 2:
            cq.serveOrder();
            break;
        case 3:
            cq.displayOrders();
            break;
        case 4:
            cout << "Exiting program." << endl;
            return 0;
        default:
            cout << "Invalid choice. Please try again." << endl;
```

```
    }
  }
}
```