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

class PizzaOrderQueue {
private:
    int front, rear, count, capacity;
    string* orders;

public:
    // Constructor
    PizzaOrderQueue(int maxOrders) {
        capacity = maxOrders;
        orders = new string[capacity];
        front = rear = -1;
        count = 0;
    }

    // Destructor
    ~PizzaOrderQueue() {
        delete[] orders;
    }

    // Function to add a new order
    bool placeOrder(const string& order) {
        if (isFull()) {
            cout << "The order queue is full. Cannot accept more orders.\n";
            return false;
        }

        // Placing the first order
        if (isEmpty()) {
            front = rear = 0;
        } else {
            // Move rear circularly
            rear = (rear + 1) % capacity;
        }

        orders[rear] = order;
        count++;
        cout << "Order placed: " << order << endl;
        return true;
    }
```

```
// Function to serve the next order
bool serveOrder() {
    if (isEmpty()) {
        cout << "No orders to serve.\n";
        return false;
    }

    cout << "Order served: " << orders[front] << endl;

    // Check if it's the last order
    if (front == rear) {
        front = rear = -1; // Reset the queue
    } else {
        // Move front circularly
        front = (front + 1) % capacity;
    }

    count--;
    return true;
}

// Function to display the current orders
void displayOrders() const {
    if (isEmpty()) {
        cout << "No orders in the queue.\n";
        return;
    }

    cout << "Current orders in the queue:\n";
    int i = front;
    while (true) {
        cout << orders[i] << (i == rear ? "\n" : " -> ");
        if (i == rear) break;
        i = (i + 1) % capacity;
    }
}

// Check if the queue is full
bool isFull() const {
    return count == capacity;
}

// Check if the queue is empty
bool isEmpty() const {
```

```
        return count == 0;
    }
};

int main() {
    int maxOrders;
    cout << "Enter the maximum number of orders the queue can hold: ";
    cin >> maxOrders;

    PizzaOrderQueue orderQueue(maxOrders);

    int choice;
    string order;
    do {
        cout << "\nPizza Order System:\n";
        cout << "1. Place Order\n";
        cout << "2. Serve Order\n";
        cout << "3. Display Orders\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter the order description: ";
                cin.ignore();
                getline(cin, order);
                orderQueue.placeOrder(order);
                break;
            case 2:
                orderQueue.serveOrder();
                break;
            case 3:
                orderQueue.displayOrders();
                break;
            case 4:
                cout << "Exiting the program.\n";
                break;
            default:
                cout << "Invalid choice. Please try again.\n";
        }
    } while (choice != 4);

    return 0;
```

```
}
```