

```

#include <iostream>
#include <string>

using namespace std;

#define MAX 100

struct Patient {
    string name;
    int priority;
};

class PriorityQueue {
private:
    Patient heap[MAX];
    int size;

    void swap(Patient &a, Patient &b) {
        Patient temp = a;
        a = b;
        b = temp;
    }

    void heapifyUp(int index) {
        while (index > 0) {
            int parent = (index - 1) / 2;
            if (heap[index].priority > heap[parent].priority) {
                swap(heap[index], heap[parent]);
                index = parent;
            } else {
                break;
            }
        }
    }

    void heapifyDown(int index) {
        int largest = index;
        int left = 2 * index + 1;
        int right = 2 * index + 2;

        if (left < size && heap[left].priority > heap[largest].priority)
            largest = left;
        if (right < size && heap[right].priority > heap[largest].priority)
            largest = right;
    }
};

```

```

        if (largest != index) {
            swap(heap[index], heap[largest]);
            heapifyDown(largest);
        }
    }
}

```

public:

```

PriorityQueue() {
    size = 0;
}

```

```

void insert(string name, int priority) {
    if (size >= MAX) {
        cout << "Queue is full!" << endl;
        return;
    }
    heap[size] = {name, priority};
    heapifyUp(size);
    size++;
}

```

```

void extractMax() {
    if (size == 0) {
        cout << "Queue is empty!" << endl;
        return;
    }
    cout << "Serving patient: " << heap[0].name << " with priority: " << heap[0].priority << endl;
    heap[0] = heap[size - 1];
    size--;
    heapifyDown(0);
}

```

```

void display() {
    if (size == 0) {
        cout << "Queue is empty!" << endl;
        return;
    }
    cout << "Patients in the queue:" << endl;
    for (int i = 0; i < size; i++) {
        cout << "Name: " << heap[i].name << " | Priority: ";
        switch (heap[i].priority) {
            case 10: cout << "Serious"; break;
            case 5: cout << "Non-serious"; break;
        }
    }
}

```

```

        case 1: cout << "Checkup"; break;
        default: cout << heap[i].priority;
    }
    cout << endl;
}
}
};

```

```

int main() {
    PriorityQueue pq;
    int choice, priority;
    string name;

    do {
        cout << "\n--- Hospital Priority Queue (Max Heap using Array) ---\n";
        cout << "1. Insert Patient\n2. Serve Patient (Highest Priority)\n3. Display Queue\n0. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter patient name: ";
                cin >> name;
                cout << "Enter Priority (10: Serious, 5: Non-serious, 1: Checkup): ";
                cin >> priority;
                pq.insert(name, priority);
                break;

            case 2:
                pq.extractMax();
                break;

            case 3:
                pq.display();
                break;

            case 0:
                cout << "Exiting...\n";
                break;

            default:
                cout << "Invalid choice!\n";
        }
    }
}

```

```
    } while (choice != 0);  
    return 0;  
}
```