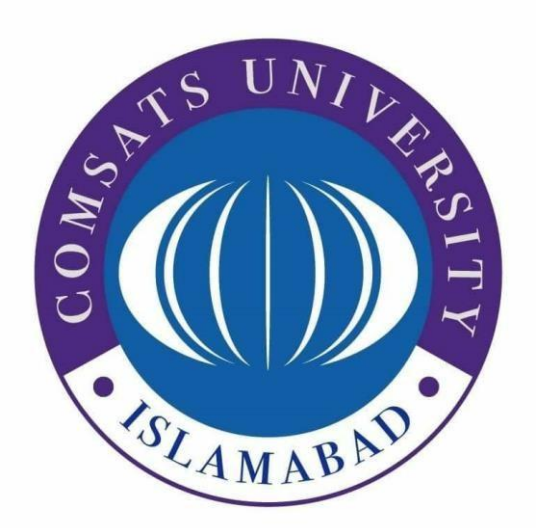

Comsats University Islamabad

Attock Campus

Name: Muhammad Talha Rafique

Reg no: SP23-BSE-028

Subject: DSA Theory

Assignment No: 02

Submitted To: Sir Muhammad Kamran

Q1:

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

struct Patient {
    string name;
    int age;
    int emergency Level; // Higher number = more serious

    Patient(string n, int a, int e) {
        name = n;
        age = a;
        emergencyLevel = e;
    }
};

// Priority Queue class using vector (linear data structure)
class PriorityQueue {
private:
    vector<Patient> patients;

public:
```

```
void addPatient(string name, int age, int emergencyLevel) {
    patients.push_back(Patient(name, age, emergencyLevel));
    sortQueue();
    cout << "\nPatient added successfully!\n";
    printQueue();
}

void sortQueue() {
    sort(patients.begin(), patients.end(), [](Patient a, Patient b) {
        return a.emergencyLevel > b.emergencyLevel;
    });
}

void treatPatient() {
    if (patients.empty()) {
        cout << "No patients to treat.\n";
        return;
    }
    cout << "\nTreating patient: " << patients[0].name
         << " (Emergency Level: " << patients[0].emergencyLevel << ")\n";
    patients.erase(patients.begin());
    printQueue();
}
```

```
    void printQueue() {
        if (patients.empty()) {
            cout << "No patients in queue.\n";
            return;
        }
        cout << "\n--- Current Treatment Queue ---\n";
        for (int i = 0; i < patients.size(); ++i) {
            cout << i + 1 << ". " << patients[i].name
                 << " (Age: " << patients[i].age
                 << ", Emergency: " << patients[i].emergencyLevel << ")\n";
        }
        cout << "-----------------------------\n";
    }
};

int main() {
    PriorityQueue pq;
    int choice;

    do {
        cout << "\n--- Hospital Emergency System ---\n";
        cout << "1. Add Patient\n";
        cout << "2. Treat Next Patient\n";
        cout << "3. Show Queue\n";
```

```
    cout << "4. Exit\n";
    cout << "Enter your choice: ";
    cin >> choice;

    if (choice == 1) {
        string name;
        int age, emergency;
        cout << "Enter patient's name: ";
        cin.ignore();
        getline(cin, name);
        cout << "Enter age: ";
        cin >> age;
        cout << "Enter emergency level (1-10): ";
        cin >> emergency;
        pq.addPatient(name, age, emergency);
    } else if (choice == 2) {
        pq.treatPatient();
    } else if (choice == 3) {
        pq.printQueue();
    } else if (choice != 4) {
        cout << "Invalid choice!\n";
    }

} while (choice != 4);
```

```
    cout << "Exiting system. Goodbye!\n";
    return 0;
}
```