The University of Azad Jammu and Kashmir

Submitted by: Aima Batool

Course: Discrete Structure & Algorithm

Roll no: 11

Submitted to: Sir Zeshan

Task: Linked List, 05

CODE:

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

struct Node {
    int patientID;
    Node* next;
    Node(int id) : patientID(id), next(nullptr) {}
};

class LinkedList {
private:
    Node* head;

public:
    LinkedList() : head(nullptr) {}

    void insertAtBeginning(int id) {
        Node* newNode = new Node(id);
        newNode->next = head;
        head = newNode;
    }
    void insertAtEnd(int id) {
        Node* newNode = new Node(id);
        if (head == nullptr) {
            head = newNode;
            return;
        }
        Node* temp = head;
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
    void displayAll() {
        if (head == nullptr) {
            cout << "No patients in the list.\n";
            return;
        }
        Node* temp = head;
        cout << "Patients: ";
        while (temp != nullptr) {
            cout << temp->patientID;
```

```cpp
            if (temp->next != nullptr) cout << " -> ";
            temp = temp->next;
        }
        cout << endl;
    }
};

int main() {
    LinkedList list;
    int choice, id;
    while (true) {
        cout << "\nMenu\n";
        cout << "1. Insert patient at end\n";
        cout << "2. Insert patient at beginning\n";
        cout << "3. Display all patients\n";
        cout << "0. Exit\n";
        cout << "Your choice: ";
        cin >> choice;
        switch (choice) {
            case 1:
                cout << "Enter patient ID: ";
                cin >> id;
                list.insertAtEnd(id);
                break;

            case 2:
                cout << "Enter emergency patient ID: ";
                cin >> id;
                list.insertAtBeginning(id);
                break;

            case 3:
                list.displayAll();
                break;

            case 0:
                return 0;

            default:
                cout << "Invalid option.\n";
        }
    }
}
```

Output

Enter choice: 1
Enter patient ID to insert at end: 101
Inserted at end.

Enter choice: 2
Enter patient ID to insert at beginning: 200
Inserted at beginning.

Enter choice: 3
Current patients (IDs): 200 -> 101