Name : Snehal Jayprakash Borji

Class : FYMCA / BATCH_A

UID_NO: 2023510008

Aim : (Implementation of link list) Perform following operations.

1. Insert
2. delete
3. Traversal

Linklist.cpp

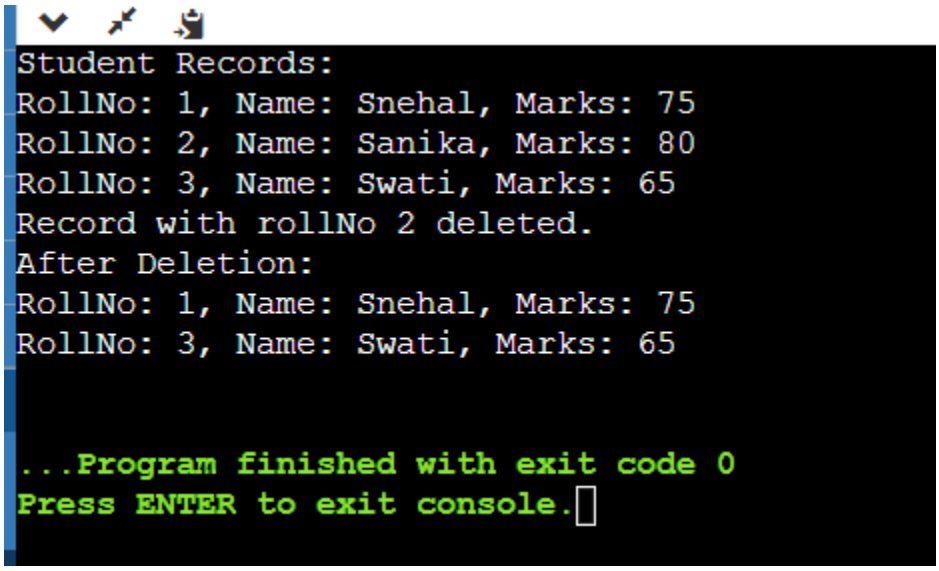
```
#include <iostream>
#include<string>
using namespace std;
struct Student {
    int rollNo;
    string name;
    float marks;
    Student* next;
};

class LinkedList {
private:
    Student* head;
public:
```

```
    LinkedList() {
 head = nullptr;
    }
void insert(int rollNo, string name, float marks) {
        Student* newNode = new Student;
        newNode->rollNo = rollNo;
        newNode->name = name;
        newNode->marks = marks;
        newNode->next = nullptr;
 if (head == nullptr || rollNo < head->rollNo) {
           newNode->next = head;
            head = newNode;
        } else {
            Student* current = head;
            while (current->next != nullptr && rollNo > current->next->rollNo) {
                current = current->next;
            }
            newNode->next = current->next;
            current->next = newNode;
        }
    }
void deleteRecord(int rollNo) {
        if (head == nullptr) {
            cout << "List is empty. Cannot delete." << endl;
            return;
```

```
    }

    if (head->rollNo == rollNo) {
            Student* temp = head;
        head = head->next;
        delete temp;
        cout << "Record with rollNo " << rollNo << " deleted." << endl;
        return;
    }

    Student* current = head;
    while (current->next != nullptr && current->next->rollNo != rollNo) {
        current = current->next;
    }

    if (current->next != nullptr) {
        Student* temp = current->next;
        current->next = current->next->next;
        delete temp;
        cout << "Record with rollNo " << rollNo << " deleted." << endl;
    } else {
        cout << "Record with rollNo " << rollNo << " not found." << endl;
    }
}
```

```
    void print() {
        Student* current = head;
        while (current != nullptr) {
            cout << "RollNo: " << current->rollNo << ", Name: " << current->name
<< ", Marks: " << current->marks << endl;
            current = current->next;
        }
    }
};
int main() {
    LinkedList list;
   list.insert(1, "Snehal", 75);
    list.insert(2, "Sanika", 80);
    list.insert(3, "Swati", 65);
cout << "Student Records:" << endl;
    list.print();
list.deleteRecord(2);
cout << "After Deletion:" << endl;
    list.print();
return 0;
}
```

OUTPUT :


Student Records:
RollNo: 1, Name: Snehal, Marks: 75
RollNo: 2, Name: Sanika, Marks: 80
RollNo: 3, Name: Swati, Marks: 65
Record with rollNo 2 deleted.
After Deletion:
RollNo: 1, Name: Snehal, Marks: 75
RollNo: 3, Name: Swati, Marks: 65
...Program finished with exit code 0
Press ENTER to exit console.