Aim:

Create a linked list with nodes having information about a student and perform

a). insert a new node at the specified position

b). delete a node with roll number of student specified

c). reversal of that linked list

Theory:

A linked list is a data structure that stores a sequence of elements.

Each element in the list is called a node, and each node has a reference to the next node in the list.

The first node in the list is called the head, and the last node in the list is called the tail.

Code:

Input:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Student {
    int rollNumber;
    char name[50];
    double grade;
    struct Student* next;
};

struct Student* createNode(int rollNumber, const char* name, double grade) {
    struct Student* newNode = (struct Student*)malloc(sizeof(struct Student));
    newNode->rollNumber = rollNumber;
    strcpy(newNode->name, name);
    newNode->grade = grade;
    newNode->next = NULL;
    return newNode;
```

```
}

void insertAtPosition(struct Student** head, int rollNumber, const char* name, double grade, int position) {
    struct Student* newNode = createNode(rollNumber, name, grade);

    if (position == 1) {
        newNode->next = *head;
        *head = newNode;
        return;
    }

    struct Student* temp = *head;
    for (int i = 1; i < position - 1 && temp != NULL; i++) {
        temp = temp->next;
    }

    if (temp == NULL) {
        printf("Position out of bounds\n");
        free(newNode);
        return;
    }

    newNode->next = temp->next;
    temp->next = newNode;
}

void deleteByRollNumber(struct Student** head, int rollNumber) {
    if (*head == NULL) {
        printf("List is empty\n");
        return;
```

```
    }

    if ((*head)->rollNumber == rollNumber) {
        struct Student* temp = *head;
        *head = (*head)->next;
        free(temp);
        return;
    }

    struct Student* temp = *head;
    while (temp->next != NULL && temp->next->rollNumber != rollNumber) {
        temp = temp->next;
    }

    if (temp->next == NULL) {
        printf("Roll number not found\n");
        return;
    }

    struct Student* nodeToDelete = temp->next;
    temp->next = temp->next->next;
    free(nodeToDelete);
}

void reverseList(struct Student** head) {
    struct Student* prev = NULL;
    struct Student* current = *head;
    struct Student* next = NULL;

    while (current != NULL) {
        next = current->next;
```

```
        current->next = prev;
        prev = current;
        current = next;
    }

    *head = prev;
}

void displayList(struct Student* head) {
    struct Student* temp = head;
    while (temp != NULL) {
        printf("Roll Number: %d, Name: %s, Grade: %f\n", temp->rollNumber, temp->name, temp->grade);
        temp = temp->next;
    }
    printf("\n");
}

int main() {
    struct Student* head = NULL;

    insertAtPosition(&head, 1, "Pratham Setia", 95.0, 1);
    insertAtPosition(&head, 2, "Parth Giri", 90.0, 2);
    insertAtPosition(&head, 3, "Swastik", 95.0, 3);

    printf("Initial Linked List:\n");
    displayList(head);

    insertAtPosition(&head, 4, "Parv Setia", 92.0, 2);
    printf("After Insertion at Position 2:\n");
    displayList(head);
```

```
    deleteByRollNumber(&head, 2);
    printf("After Deletion of Roll Number 2:\n");
    displayList(head);

    reverseList(&head);
    printf("After Reversal of the Linked List:\n");
    displayList(head);

    return 0;
}
```

Output:

```
Initial Linked List:
Roll Number: 1, Name: Daksh, Grade: 85.00
Roll Number: 2, Name: Lalit, Grade: 90.00
Roll Number: 3, Name: Anish, Grade: 95.00

After Insertion at Position 2:
Roll Number: 1, Name: Daksh, Grade: 85.00
Roll Number: 4, Name: Aryan, Grade: 92.00
Roll Number: 2, Name: Lalit, Grade: 90.00
Roll Number: 3, Name: Anish, Grade: 95.00

After Deletion of Roll Number 2:
Roll Number: 1, Name: Daksh, Grade: 85.00
Roll Number: 4, Name: Aryan, Grade: 92.00
Roll Number: 3, Name: Anish, Grade: 95.00

After Reversal of the Linked List:
Roll Number: 3, Name: Anish, Grade: 95.00
```

Roll Number: 4, Name: Aryan, Grade: 92.00

Roll Number: 1, Name: Daksh, Grade: 85.00