Rajalakshmi Engineering College

Name: Naren S
Email: 241901066@rajalakshmi.edu.in
Roll no: 241901066
Phone: 6382463115
Branch: REC
Department: l CSE (CS) FA
Batch: 2028
Degree: B.E - CSE (CS)

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Imagine you are tasked with developing a simple GPA management system using a singly linked list. The system allows users to input student GPA values, insertion should happen at the front of the linked list, delete record by position, and display the updated list of student GPAs.

Input Format

The first line of input contains an integer n, representing the number of students.

The next n lines contain a single floating-point value representing the GPA of each student.

The last line contains an integer position, indicating the position at which a student record should be deleted. Position starts from 1.

After deleting the data in the given position, display the output in the format "GPA: " followed by the GPA value, rounded off to one decimal place.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
3.8
3.2
3.5
4.1
2
Output: GPA: 4.1
GPA: 3.2
GPA: 3.8

Answer

```c
// You are using GCC
#include <stdio.h>
#include <stdlib.h>

// Define the node structure
struct Node {
    float gpa;
    struct Node* next;
};

// Insert GPA at the front of the linked list
void insertFront(struct Node** head, float gpa) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->gpa = gpa;
    newNode->next = *head;
    *head = newNode;
}

// Delete node at a specific position (1-based)
void deleteAtPosition(struct Node** head, int position) {
```

```c
    if (*head == NULL) return;

    struct Node* temp = *head;

    // Delete head node
    if (position == 1) {
        *head = temp->next;
        free(temp);
        return;
    }

    // Traverse to the node before the one to delete
    for (int i = 1; temp != NULL && i < position - 1; i++) {
        temp = temp->next;
    }

    // If position is more than number of nodes
    if (temp == NULL || temp->next == NULL) return;

    struct Node* toDelete = temp->next;
    temp->next = toDelete->next;
    free(toDelete);
}

// Display the GPA list
void displayList(struct Node* head) {
    struct Node* current = head;
    while (current != NULL) {
        printf("GPA: %.1f", current->gpa);
        if (current->next != NULL) printf(" ");
        current = current->next;
    }
}

int main() {
    int n, pos;
    float gpa;
    struct Node* head = NULL;

    scanf("%d", &n);

    // Read and insert GPAs at the front
```

```c
    for (int i = 0; i < n; i++) {
        scanf("%f", &gpa);
        insertFront(&head, gpa);
    }

    scanf("%d", &pos);

    deleteAtPosition(&head, pos);
    displayList(head);

    return 0;
}
```

Status : Correct   *Marks : 10/10*