Rajalakshmi Engineering College

Name: Adhithya varun
Email: 240801008@rajalakshmi.edu.in
Roll no: 240801008
Phone: null
Branch: REC
Department: I ECE FA
Batch: 2028
Degree: B.E - ECE

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.

Output Format

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

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

struct Node{
   float gpa;
   struct Node* next;
};

struct Node* createNode(float gpa){
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->gpa=gpa;
   newNode->next=NULL;
   return newNode;
}

struct Node* deleteSTudentAtPosition(struct Node* head,int position){
   if (head == NULL){
     return NULL;
   }
```

```
  if (position == 1){
     struct Node* temp = head;
     head = head->next;
     free(temp);

  }else{
  struct Node* prev = NULL;
  struct Node* current = head;
  int currentPosition = 1;

  while(currentPosition<position && current != NULL){
      prev = current;
      current=current->next;
      currentPosition++;
  }

  if(current != NULL){
      prev->next = current->next;
      free(current);
  }

 }
 return head;
}

void printStudentList(struct Node* head){
   struct Node* current=head;
   while (current != NULL){
      printf("GPA: %.1f\n", current->gpa);
      current = current->next;
   }
}

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

   int n;
   scanf("%d",&n);

   for(int i=0;i<n;++i){
      float gpa;
      scanf("%f",&gpa);

      if(head == NULL){
          head = createNode(gpa);
      }else{
          struct Node* newNode=createNode(gpa);
          newNode->next=head;
          head=newNode;
      }







  }
  int position;
  scanf("%d ",&position);

  head=deleteSTudentAtPosition(head,position);

  printStudentList(head);

  return 0;
}
```

Status : Correct ***Marks : 10/10***