Rajalakshmi Engineering College

Name: Karthikeyan M
Email: 240801150@rajalakshmi.edu.in
Roll no: 2116240801150
Phone: 8056008890
Branch: REC
Department: I ECE FB
Batch: 2028
Degree: B.E - ECE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 2

Attempt : 2
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Arun is learning about data structures and algorithms. He needs your help in solving a specific problem related to a singly linked list.

Your task is to implement a program to delete a node at a given position. If the position is valid, the program should perform the deletion; otherwise, it should display an appropriate message.

Input Format

The first line of input consists of an integer N, representing the number of elements in the linked list.

The second line consists of N space-separated elements of the linked list.

The third line consists of an integer x, representing the position to delete.

Position starts from 1.

Output Format

The output prints space-separated integers, representing the updated linked list after deleting the element at the given position.

If the position is not valid, print "Invalid position. Deletion not possible."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
8 2 3 1 7
2
Output: 8 3 1 7

Answer

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

void insert(int);
void display_List();
void deleteNode(int);

struct node {
   int data;
   struct node* next;
} *head = NULL, *tail = NULL;

void insert(int data) {
   struct node* newNode = (struct node*)malloc(sizeof(struct node));
   if (newNode == NULL) {
      perror("Failed to allocate memory for new node");
      exit(EXIT_FAILURE);
   }
   newNode->data = data;
   newNode->next = NULL;

   if (head == NULL) {
        head = newNode;
        tail = newNode;
    } else {
        tail->next = newNode;
        tail = newNode;
    }
}

void display_list() {
    struct node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    printf("\n");
}

void deleteNode(int position) {
    if (head == NULL) {
        printf("Invalid position. Deletion not possible.\n");
        return;
    }

    struct node* temp = head;
    struct node* prev = NULL;
    int count = 1;

    if (position == 1) {
        head = temp->next;
        if (head == NULL) {
            tail = NULL;
        }
        free(temp);
        display_list();
        return;
    }

    while (temp != NULL && count < position) {
        prev = temp;
        temp = temp->next;
        count++;
    }
    if (temp == NULL) {
        printf("Invalid position. Deletion not possible.\n");
        return;
    }

    prev->next = temp->next;

    if (temp == tail) {
        tail = prev;
    }

    free(temp);
    display_list();
}



int main() {
    int num_elements, element, pos_to_delete;

    scanf("%d", &num_elements);

    for (int i = 0; i < num_elements; i++) {
        scanf("%d", &element);
        insert(element);
    }

    scanf("%d", &pos_to_delete);

    deleteNode(pos_to_delete);

    return 0;
}
```

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