Rajalakshmi Engineering College

Name: Divya darshini S
Email: 241501051@rajalakshmi.edu.in
Roll no: 241501051
Phone: 6383045036
Branch: REC
Department: I AIML FA
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

As part of a programming assignment in a data structures course, students are required to create a program to construct a singly linked list by inserting elements at the beginning.

You are an evaluator of the course and guide the students to complete the task.

Input Format

The first line of input consists of an integer N, which is the number of elements.

The second line consists of N space-separated integers.

Output Format

The output prints the singly linked list elements, after inserting them at the beginning.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
78 89 34 51 67

Output: 67 51 34 89 78

Answer

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

struct Node {
   int data;
   struct Node* next;
};
```

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

// Define the structure of a node
struct Node {
   int data;
   struct Node* next;
};

// Function to insert a new node at the beginning
void insertAtBeginning(struct Node** head_ref, int new_data) {
   // Allocate memory for new node
   struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
   new_node->data = new_data;
   new_node->next = *head_ref; // Point to old head
   *head_ref = new_node; // Move head to point to new node
}

// Function to print the linked list
void printList(struct Node* node) {
```

```
    while (node != NULL) {
        printf("%d ", node->data);
        node = node->next;
    }
}

int main() {
    int N, i, val;
    struct Node* head = NULL;

    // Read number of elements
    scanf("%d", &N);

    // Check constraints
    if (N < 1 || N > 10) {
        printf("Invalid number of elements.\n");
        return 1;
    }

    // Read N elements and insert them at the beginning
    for (i = 0; i < N; i++) {
        scanf("%d", &val);
        if (val < 1 || val > 100) {
            printf("Invalid element value.\n");
            return 1;
        }
        insertAtBeginning(&head, val);
    }

    // Print the final linked list
    printList(head);

    return 0;
}


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

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

    for (int i = 0; i < n; i++) {
```

```
        int activity;
        scanf("%d", &activity);
        insertAtFront(&head, activity);
    }

    printList(head);
    struct Node* current = head;
    while (current != NULL) {
        struct Node* temp = current;
        current = current->next;
        free(temp);
    }

    return 0;
}
```

Status : Wrong **Marks : 0/10**