Rajalakshmi Engineering College

Name: Jhanani shree
Email: 240701215@rajalakshmi.edu.in
Roll no: 240701215
Phone: 7373333511
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ravi is developing a student registration system for a college. To efficiently store and manage the student IDs, he decides to implement a doubly linked list where each node represents a student's ID.

In this system, each student's ID is stored sequentially, and the system needs to display all registered student IDs in the order they were entered.

Implement a program that creates a doubly linked list, inserts student IDs, and displays them in the same order.

Input Format

The first line contains an integer N the number of student IDs.

The second line contains N space-separated integers representing the student IDs.

Output Format

The output should display the single line containing N space-separated integers representing the student IDs stored in the doubly linked list.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
10 20 30 40 50
Output: 10 20 30 40 50

Answer

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

struct Node {
    int id;
    struct Node* next;
    struct Node* prev;
};

void insert_at_end(struct Node** head, int id) {
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
    new_node->id = id;
    new_node->next = NULL;

    if (*head == NULL) {
        new_node->prev = NULL;
        *head = new_node;
    } else {
        struct Node* temp = *head;

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

```
        temp->next = new_node;
        new_node->prev = temp;
    }
}

void display(struct Node* head) {
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d ", temp->id);
        temp = temp->next;
    }
    printf("\n");
}

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

int main() {
    int N;
    scanf("%d", &N);
    struct Node* head = NULL;

    for (int i = 0; i < N; i++) {
        int id;
        scanf("%d", &id);
        insert_at_end(&head, id);
    }

    display(head);

    free_list(head);

    return 0;
}
```

Status : Correct *Marks : 10/10*