Rajalakshmi Engineering College

Name: Praveen Kumar
Email: 240801247@rajalakshmi.edu.in
Roll no: 240801247
Phone: 7550385160
Branch: REC
Department: I ECE AF
Batch: 2028
Degree: B.E - ECE

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

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

typedef struct ID {
    int value;
    struct ID* next;
    struct ID* prev;
}Node;

void insertNode(Node** head, int value) {
    Node* newnode = (Node*) malloc(sizeof(Node));
    newnode->value = value;
    newnode->next = NULL;
    newnode->prev = NULL;

    if(*head == NULL) {
        *head = newnode;
        return;
    }
    Node* temp = *head;
    while(temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newnode;
    newnode->prev = temp;
}

void traverse(Node* head) {
    while(head != NULL) {
        printf("%d ", head->value);
        head = head->next;
    }
    printf("\n");
}

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

    for(int i = 0; i < n; i++) {
        scanf("%d", &m);
        insertNode(&head, m);
    }
    traverse(head);

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

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