Rajalakshmi Engineering College

Name: Gowtham P
Email: 241501060@rajalakshmi.edu.in
Roll no: 241501060
Phone: 8838517270
Branch: REC
Department: I AI & ML FA
Batch: 2028
Degree: B.E - AI & ML

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>

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


struct Node* InsertNode(struct Node* head,struct Node* newNode){
   if(head == NULL){
      head = newNode;

      newNode -> prev = NULL;
      newNode -> next = NULL;
      return head;
   }
   struct Node* temp = head;
   while(temp -> next != NULL){
      temp = temp -> next;
   }
   newNode -> prev = temp;
   temp -> next = newNode;
```

```
    newNode -> next = NULL;
    return newNode;


}
int main(){
    int n;
    struct Node* head = NULL;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        struct Node* newNode;
        newNode = (struct Node*) malloc (sizeof(struct Node));
        scanf("%d ",&newNode -> data);
        InsertNode(head,newNode);
        printf("%d ",newNode -> data);

    }
}
```

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