Rajalakshmi Engineering College

Name: Kirithick R
Email: 240701627@rajalakshmi.edu.in
Roll no: 2116240701627
Phone: 9952595005
Branch: REC
Department: I CSE FF
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

```
#include<stdio.h>
#include<stdlib.h>
struct Node{
   int Stu_Id;
   struct Node* prev;
   struct Node* next;
};
struct Node* list=NULL;
void traverse(struct Node* List){
   if(List!=NULL){
      struct Node* position;
      position=List;
      while(position->next!=NULL){
         printf("%d ",position->Stu_Id);
         position=position->next;
      }
   }
}
int main(){
   int N;
   scanf("%d",&N);
   for(int i=0;i<=N;i++){
      struct Node* newnode=(struct Node*)malloc(sizeof(struct Node));
      scanf("%d ",&newnode->Stu_Id);
```

```
        newnode->next=NULL;
        newnode->prev=NULL;
        struct Node* position;
        if(list==NULL){
          list=newnode;
        }
        else{
            position=list;
            while(position->next!=NULL){
                position=position->next;
            }
            position->next=newnode;
            newnode->prev=position;
        }
    }
    traverse(list);
}
```

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