Rajalakshmi Engineering College

Name: Sarvesh S
Email: 240701479@rajalakshmi.edu.in
Roll no: 240701479
Phone: 9361488694
Branch: REC
Department: I CSE FE
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 element;
   struct node *next;
   struct node *prev;
};
typedef struct node Node;

void insertend(Node *List,int e){
   Node *newnode=(Node*)malloc(sizeof(Node));
   Node *position;
   newnode->element=e;
   newnode->next=NULL;
   if(List->next==NULL){
      newnode->prev=List;
      List->next=newnode;
   }
   else{
      position=List;
      while(position->next!=NULL){
         position=position->next;
      }
      newnode->prev=position;
```

```
        position->next=newnode;
    }
}

void traverse(Node *List){
    Node *position=List->next;
    while(position!=NULL){
        printf("%d ",position->element);
        position=position->next;
    }
    printf("\n");
}
int main(){
    Node *List=(Node*)malloc(sizeof(Node));
    List->next=NULL;
    int n,e;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf("%d",&e);
        insertend(List,e);
    }
    traverse(List);
    return 0;
}
```

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