Rajalakshmi Engineering College

Name: Aslam Churihar
Email: 240701061@rajalakshmi.edu.in
Roll no: 240701061
Phone: 8188836253
Branch: REC
Department: I CSE AG
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 : 2
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>

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

Node *head=NULL;

Node *createNode(int data){
   Node *newNode=(Node*)malloc(sizeof(Node));
   if(newNode==NULL){
      printf("Memory allocation failed!");
      exit(0);
   }
   newNode->id=data;
   newNode->next=NULL;
   newNode->prev=NULL;

   return newNode;
}
```

```
void insertEnd(Node **head,int data){
    Node *newNode=createNode(data);

    if((*head)==NULL){
        *head=newNode;
        return;
    }

    Node *ptr=*head;
    while(ptr->next!=NULL){
        ptr=ptr->next;
    }
    ptr->next=newNode;
    newNode->prev=ptr;
}

void display(Node *head){
    if(head==NULL){
        printf("Linked List is empty:");
        exit(0);
    }
    Node *ptr=head;
    while(ptr!=NULL){
        printf("%d ",ptr->id);
        ptr=ptr->next;
    }
}

int main(){
    int n;
    scanf("%d",&n);

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

    display(head);

    return 0;
```

```
}
```

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