Rajalakshmi Engineering College

Name: Subhalakshmi M
Email: 240701539@rajalakshmi.edu.in
Roll no: 240701539
Phone: 6379032776
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 info;
   struct node* prev;
   struct node* next;
};

struct node* head = NULL;

void insertAtEnd(int data) {
   struct node* newNode = (struct node*)malloc(sizeof(struct node));
   newNode->info = data;
   newNode->prev = NULL;
   newNode->next = NULL;

   if (head == NULL) {
      head = newNode;
   } else {
      struct node* temp = head;
      while (temp->next != NULL) {
         temp = temp->next;
```

```
        }
        temp->next = newNode;
        newNode->prev = temp;
    }
}


void traverse() {
    struct node* temp = head;
    while (temp != NULL) {
        printf("%d ", temp->info);
        temp = temp->next;
    }
}

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

    for (int i = 0; i < n; i++) {
        scanf("%d", &data);
        insertAtEnd(data);
    }

    traverse();

    return 0;
}
```

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