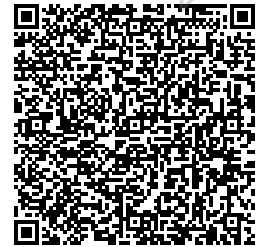


# Rajalakshmi Engineering College

Name: SUBASH R  
Email: 240701538@rajalakshmi.edu.in  
Roll no: 240701538  
Phone: 9150202710  
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>
```

```
typedef struct node{  
    struct node*prev;  
    int ele;  
    struct node*next;  
}node;
```

```
void insertend(node*list,int e){  
    node*position=list;  
    while(position->next!=NULL)  
        position=position->next;  
    node*newnode=(node*)malloc(sizeof(node));  
    newnode->ele=e;  
    newnode->prev=position;  
    newnode->next=NULL;  
    position->next=newnode;  
}
```

```
void traverse(node*list){  
    node*position=list->next;  
    while(position!=NULL){  
        printf("%d ",position->ele);
```

```
        position=position->next;
    }
}
int main(){
    node*list=(node*)malloc(sizeof(node));
    list->next=NULL;
    int n;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        int e;
        scanf("%d",&e);
        insertend(list,e);
    }
    traverse(list);
}
```

**Status :** Correct

**Marks : 10/10**