# Rajalakshmi Engineering College

Name: Aishwarya R
Email: 241501011@rajalakshmi.edu.in
Roll no: 241501011
Phone: null
Branch: REC
Department: I AI & ML FA
Batch: 2028
Degree: B.E - AI & ML

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***

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>

// Define a node structure for the doubly linked list
struct Node {
    int data;
    struct Node* prev;
    struct Node* next;
};
struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->prev = NULL;
    newNode->next = NULL;
    return newNode;
}

int main() {
    int N, i, value;
    scanf("%d", &N);

    struct Node* head = NULL;
    struct Node* tail = NULL;
```

```
    for (i = 0; i < N; i++) {
        scanf("%d", &value);
        struct Node* newNode = createNode(value);
        if (head == NULL) {
            head = newNode;
            tail = newNode;
        }
        else
        {
            tail->next = newNode;
            newNode->prev = tail;
            tail = newNode;
        }
    }

    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d", temp->data);
        if (temp->next != NULL) {
            printf(" ");}
        temp=temp->next;
    }
}
```

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