

# Rajalakshmi Engineering College

Name: vinnarasan j  
Email: 241501245@rajalakshmi.edu.in  
Roll no: 241501245  
Phone: 7305999373  
Branch: REC  
Department: I AI & ML FC  
Batch: 2028  
Degree: B.E - AI & ML

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 1\_COD\_Question 6

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

John is tasked with creating a program to manage student roll numbers using a singly linked list.

Write a program for John that accepts students' roll numbers, inserts them at the end of the linked list, and displays the numbers.

##### ***Input Format***

The first line of input consists of an integer N, representing the number of students.

The second line consists of N space-separated integers, representing the roll numbers of students.

##### ***Output Format***

The output prints the space-separated integers singly linked list, after inserting the roll numbers of students at the end.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 5

23 85 47 62 31

Output: 23 85 47 62 31

### **Answer**

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int roll_number;
    struct Node* next;
};

struct Node* insertEnd(struct Node* head, int new_roll) {
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
    if (new_node == NULL) {
        printf("Memory allocation failed\n");
        exit(EXIT_FAILURE);
    }
    new_node->roll_number = new_roll;
    new_node->next = NULL;

    if (head == NULL) {
        return new_node;
    }

    struct Node* current = head;
    while (current->next != NULL) {
        current = current->next;
    }
    current->next = new_node;
    return head;
}

void displayList(struct Node* head) {
    struct Node* current = head;
```

```

while (current != NULL) {
    printf("%d ", current->roll_number);
    current = current->next;
}
printf("\n");
}

void freeList(struct Node* head) {
    struct Node* current = head;
    struct Node* next;
    while (current != NULL) {
        next = current->next;
        free(current);
        current = next;
    }
}

int main() {
    int n, roll;
    struct Node* head = NULL;

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

    displayList(head);
    freeList(head);

    return 0;
}

```

**Status :** Correct

**Marks :** 10/10