

# Rajalakshmi Engineering College

Name: Dhyanesh M  
Email: 241801055@rajalakshmi.edu.in  
Roll no:  
Phone: 6380972627  
Branch: REC  
Department: I AI & DS FB  
Batch: 2028  
Degree: B.E - AI & DS

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 1\_COD\_Question 4

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

As part of a programming assignment in a data structures course, students are required to create a program to construct a singly linked list by inserting elements at the beginning.

You are an evaluator of the course and guide the students to complete the task.

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

The first line of input consists of an integer N, which is the number of elements.

The second line consists of N space-separated integers.

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

The output prints the singly linked list elements, after inserting them at the beginning.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 5

78 89 34 51 67

Output: 67 51 34 89 78

### ***Answer***

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct Node {  
    int data;  
    struct Node* next;  
};
```

```
// You are using GCC
```

```
void insertAtFront(struct Node**head,int data){  
    struct Node*newnode=(struct Node*)malloc(sizeof(struct Node));  
    if(newnode==NULL){  
        printf("Memory allocation failed\n");  
        return;  
    }  
    newnode->data=data;  
    newnode->next=*head;  
    *head=newnode;  
}  
void printList(struct Node*head){  
    struct Node*current=head;  
    while(current!=NULL){  
        printf("%d ",current->data);  
        current=current->next;  
    }  
    printf("\n");  
}
```

```
int main(){
    struct Node* head = NULL;

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

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

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

    return 0;
}
```

**Status :** Correct

**Marks :** 10/10