

# Rajalakshmi Engineering College

Name: ROHITH KUMAR S  
Email: 240701436@rajalakshmi.edu.in  
Roll no: 240701436  
Phone: 7603815548  
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 3

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Bob is tasked with developing a company's employee record management system. The system needs to maintain a list of employee records using a doubly linked list. Each employee is represented by a unique integer ID.

Help Bob to complete a program that adds employee records at the front, traverses the list, and prints the same for each addition of employees to the list.

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

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

The second line consists of N space-separated integers, representing the employee IDs.

### **Output Format**

For each employee ID, the program prints "Node Inserted" followed by the current state of the doubly linked list in the next line, with the data values of each node separated by spaces.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 4

101 102 103 104

Output: Node Inserted

101

Node Inserted

102 101

Node Inserted

103 102 101

Node Inserted

104 103 102 101

### **Answer**

```
#include <iostream>
using namespace std;
```

```
struct node {
    int info;
    struct node* prev, * next;
};
```

```
struct node* start = NULL;
```

```
typedef struct node Node;
Node dummy={0,NULL,NULL};
```

```
Node *List=&dummy;
```

```
void traverse() {
    Node *pos=List->next;
    printf("Node Inserted\n");
    while(pos!=NULL){
```

```
        printf("%d ",pos->info);
```

```

    pos=pos->next;
}
}

void insertAtFront(int data) {
    Node *newnode=(Node*)malloc(sizeof(Node));
    newnode->info=data;
    if(List->next==NULL){
        newnode->next=NULL;
    }else{
        newnode->next=List->next;
        List->next->prev=newnode;
    }
    List->next=newnode;
    newnode->prev=List;
}

```

```

int main() {
    int n, data;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> data;
        insertAtFront(data);
        traverse();
    }
    return 0;
}

```

**Status :** Correct

**Marks :** 10/10