

# Rajalakshmi Engineering College

Name: Kanija Fathima J  
Email: 240701226@rajalakshmi.edu.in  
Roll no: 240701226  
Phone: 7904195258  
Branch: REC  
Department: I CSE AH  
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 : 0

#### 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;
```

```
void DListcreation(int n) {
    int num, i;
    struct node *newnode;
    if (n >= 1) {
        stnode = (struct node *)malloc(sizeof(struct node));
        scanf("%d", &num);
        stnode->num = num;
        stnode->preptr = NULL;
```

```

    stnode->nextptr = NULL;
    ennode = stnode;
    for (i = 2; i <= n; i++) {
        newnode = (struct node *)malloc(sizeof(struct node));
        scanf("%d", &num);
        newnode->num = num;
        newnode->preptr = ennode;
        newnode->nextptr = NULL;
        ennode->nextptr = newnode;
        ennode = newnode;
    }
}
}

```

```

void DListDeleteAnyNode(int pos) {
    struct node *current;
    int i;
    if (pos < 1) {
        printf("Invalid position. Try again.\n");
    } else {
        current = stnode;
        for (i = 1; i < pos && current != NULL; i++) {
            current = current->nextptr;
        }
        if (current == NULL) {
            printf("Invalid position. Try again.\n");
        } else {
            if (current->preptr != NULL) {
                current->preptr->nextptr = current->nextptr;
            } else {
                stnode = current->nextptr;
            }
            if (current->nextptr != NULL) {
                current->nextptr->preptr = current->preptr;
            }
            free(current);
        }
    }
}

```

```

void displayDList(int m) {
    struct node *current;

```

```
int i = 1;
if (m == 1) {
    printf("Data entered in the list:\n");
} else {
    printf("After deletion the new list:\n");
}
current = stnode;
while (current != NULL) {
    printf(" node %d : %d\n", i, current->num);
    current = current->nextptr;
    i++;
}
}
```

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

**Status : Wrong**

**Marks : 0/10**