

# Rajalakshmi Engineering College

Name: Rajeshwar s  
Email: 240701413@rajalakshmi.edu.in  
Roll no: 2116240701413  
Phone: 9003785151  
Branch: REC  
Department: I CSE FD  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 2\_COD\_Question 5

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Ashwin is tasked with developing a simple application to manage a list of items in a shop inventory using a doubly linked list. Each item in the inventory has a unique identification number. The application should allow users to perform the following operations:

Create a List of Items: Initialize the inventory with a given number of items. Each item will be assigned a unique number provided by the user and insert the elements at end of the list.

Delete an Item: Remove an item from the inventory at a specific position.

Display the Inventory: Show the list of items before and after deletion.

If the position provided for deletion is invalid (e.g., out of range), it should

display an error message.

### ***Input Format***

The first line contains an integer  $n$ , representing the number of items to be initially entered into the inventory.

The second line contains  $n$  integers, each representing the unique identification number of an item separated by spaces.

The third line contains an integer  $p$ , representing the position of the item to be deleted from the inventory.

### ***Output Format***

The first line of output prints "Data entered in the list:" followed by the data values of each node in the doubly linked list before deletion.

If  $p$  is an invalid position, the output prints "Invalid position. Try again."

If  $p$  is a valid position, the output prints "After deletion the new list:" followed by the data values of each node in the doubly linked list after deletion.

Refer to the sample output for the formatting specifications.

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

Input: 4

1 2 3 4

5

Output: Data entered in the list:

node 1 : 1

node 2 : 2

node 3 : 3

node 4 : 4

Invalid position. Try again.

### ***Answer***

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

```
void DListcreation(int n) {  
    int data;
```

```

struct node*fnnode,*tmp;
if(n>=1){
    stnode=(struct node*)malloc(sizeof(struct node));
    scanf("%d",&data);
    stnode->num=data;
    stnode->preptr=NULL;
    stnode->nextptr=NULL;
    ennode=stnode;
    for(int i=2;i<=n;i++){
        fnnode=(struct node*)malloc(sizeof(struct node));
        scanf("%d",&data);
        fnnode->num=data;
        fnnode->preptr=ennode;
        fnnode->nextptr=NULL;
        ennode->nextptr=fnnode;
        ennode=fnnode;
    }
}
}

```

```

void DListDeleteAnyNode(int pos) {
    struct node*curnode;
    curnode=stnode;
    for(int i=1;i<pos&& curnode!=NULL;i++){
        curnode=curnode->nextptr;
    }
    if(curnode!=NULL){
        if(curnode->preptr!=NULL)
            curnode->preptr->nextptr=curnode->nextptr;
        else
            stnode=curnode->nextptr;
        if(curnode->nextptr!=NULL)
            curnode->nextptr->preptr=curnode->preptr;
        free(curnode);
    }
}

```

```

void DListDeleteFirstNode() {
    if(stnode!=NULL){
        struct node*tmp=stnode;
        stnode=stnode->nextptr;
    }
}

```

```

        if(stnode!=NULL){
            stnode->preptr=NULL;
        }
        free(tmp);
    }
}

```

```

void DListDeleteLastNode() {
    if(enode!=NULL){
        struct node*tmp=enode;
        enode=enode->preptr;
        if(enode!=NULL)
            enode->nextptr=NULL;
        else
            stnode=NULL;
        free(tmp);
    }
}

```

```

void displayDList(int m) {
    struct node*tmp;
    int i=1;
    if(stnode==NULL){
        printf("Invalid position. Try again.");
    }
    else{
        tmp=stnode;
        if(m==1)
            printf("Data entered in the list: \n");
        else
            printf("After deletion the new list: \n");
        while(tmp!=NULL){
            printf("node %d : %d\n",i,tmp->num);
            i++;
            tmp=tmp->nextptr;
        }
        printf("\n");
    }
}

```

**Status :** Correct

**Marks :** 10/10