

# Rajalakshmi Engineering College

Name: santhosh kumar  
Email: 240801303@rajalakshmi.edu.in  
Roll no: 240801303  
Phone: 7904117179  
Branch: REC  
Department: I ECE AF  
Batch: 2028  
Degree: B.E - ECE

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) {  
    for(int i=0;i<n;i++){
```

```

    struct node *temp=(struct node*)malloc(sizeof(struct node));
    int q;
    scanf("%d",&q);
    temp->num=q;
    temp->nextptr=NULL;
    if(ennode!=NULL){
        ennode->nextptr=temp;
        temp->preptr=ennode;
    }ennode=temp;
    if(stnode==NULL){
        stnode=temp;
        temp->preptr=NULL;
    }
}
}

```

```

void DListDeleteAnyNode(int pos) {
    struct node *temp=stnode;
    struct node *temp2=NULL;
    if(pos==1){
        DListDeleteFirstNode();
        return;
    }
    struct node *last=ennode;
    int count=1;
    while(last!=NULL&&last->nextptr!=NULL){
        last=last->nextptr;
        count++;
    }
    if(pos==count){
        DListDeleteLastNode();
        return;
    }
    while(pos!=1&&temp!=NULL){
        temp=temp->nextptr;
        pos--;
    }
    if(temp==NULL){
        printf("Invalid position");
        return;
    }
    temp2=temp->preptr;

```

```
temp2->nextptr=temp->nextptr;
if(temp->nextptr!=NULL){
temp->nextptr->preptr=temp2;}
free(temp);
temp=NULL;
}
```

```
void DListDeleteFirstNode() {
    struct node *ptr=stnode;
    stnode=stnode->nextptr;
    if(stnode!=NULL){
        stnode->preptr=NULL;
    }
    free(ptr);
    ptr=NULL;
}
```

```
void DListDeleteLastNode() {
    struct node *temp=enode;
    enode=enode->preptr;
    enode->nextptr=NULL;
    free(temp);
    temp=NULL;
}
```

```
void displayDList(int m) {
    struct node *ptr=stnode;
    int count=1;
    if(m==1){
        printf("Data entered in the list:\n");
        while(ptr!=NULL){
            printf("node %d:%d\n",count,ptr->num);
            ptr=ptr->nextptr;
            count++;
        }
    }else if(m==2){
        printf("After deletion the new list:\n");
        ptr=stnode;
        while(ptr!=NULL){
            printf("node %d: %d\n",count,ptr->num);
            ptr=ptr->nextptr;
        }
    }
}
```

```
240801303 count++;  
240801303 }  
240801303 }  
240801303 }
```

**Status :** Correct

**Marks :** 10/10