

# Rajalakshmi Engineering College

Name: syed mohammed hussain  
Email: 241801288@rajalakshmi.edu.in  
Roll no: 241801288  
Phone: 9363281289  
Branch: REC  
Department: I AI & DS FD  
Batch: 2028  
Degree: B.E - AI & DS

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
```

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

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

```
typedef struct node
{
    int data;
    struct node* next;
    struct node* prev;
}Node;
void insertAtend(Node** head,int e)
```

```
{
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode -> data = e;
    newNode -> next = NULL;
    newNode -> prev = NULL;
    if(*head == NULL)
    {
        *head = newNode;
        return;
    }
    Node* temp = *head;
    while(temp->next != NULL)
    {
        temp = temp -> next;
    }
    temp -> next = newNode;
    newNode -> prev = temp;
}
```

```
void Traverse(Node* head)
{
    Node* temp = head;
    int i=1;
    while(temp != NULL )
    {
        printf("node %d : %d\n",i,temp -> data);
        temp = temp -> next;
        i++;
    }
}
```

```
void deletionAtmid(Node** head,int p)
{
    Node* temp = *head;
    int i=1;
    while(temp != NULL && i<p)
    {
```

```

        temp = temp -> next;
        i++;
    }
    if(temp == *head)
    {
        *head = temp -> next;
    }
    if(temp -> next != NULL)
    {
        temp -> next -> prev = temp -> prev;
    }
    if(temp -> prev != NULL)
    {
        temp -> prev -> next = temp -> next;
    }
    free(temp);
}
int main()
{
    Node* head = NULL;
    int n,e;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&e);
        insertAtend(&head,e);
    }
    int p;
    scanf("%d",&p);
    printf("Data entered in the list:\n");
    Traverse(head);
    if(p>n)
    {
        printf("Invalid position. Try again.");
        return 0;
    }
    deletionAtmid(&head,p);
    printf("After deletion the new list:\n");
    Traverse(head);
}

```

**Status :** Correct

**Marks :** 10/10