**Deletion and Reverse in Linked List**

**Easy**

Given a Circular Linked List of size **N**. The task is to delete the given node (excluding the first and last node) in the circular linked list and then print the reverse of the circular linked list.

**Example 1:**

**Input:**

5

2 5 7 8 10

8

**Output:**

10 7 5 2

**Explanation:**

After deleting 8 from the given circular linked

list, it has elements as 2, 5, 7, 10. Now,

reversing this list will result in 10, 7, 5, 2.

**Example 2:**

**Input:**

4

1 7 8 10

8

**Output:**

10 7 1

**Explanation:**

After deleting 8 from the given circular linked

list, it has elements as 1, 7,10. Now, reversing

this list will result in 10, 7, 1.

**Constraints:**  
1 <= T <= 100  
1 <= n <= 100  
1 <= data <= 100

**Company Tags**

[**Intuit**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Intuit)

//{ Driver Code Starts

import java.util.Scanner;

import java.io.\*;

class Node

{

int data;

Node next;

Node(int d)

{

data=d;

next=null;

}

}

class CodingMaxima1{

public static void printList(Node head)

{

Node start=head;

System.out.print(head.data+" ");

head=head.next;

while(head!=start)

{

System.out.print(head.data+" ");

head=head.next;

}

System.out.println();

}

public static void main (String[] args) {

Scanner sc=new Scanner(System.in);

int t=sc.nextInt();

while(t-->0)

{

int n=sc.nextInt();

Node start=null,cur=null;

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

{

int a=sc.nextInt();

Node ptr=new Node(a);

if(start==null)

{

start=ptr;

cur=ptr;

}

else

{

cur.next=ptr;

cur=ptr;

}

}

int delNode=sc.nextInt();

cur.next=start;

CodingMaxima obj=new CodingMaxima ();

Node head=obj.deleteNode(start,delNode);

start=obj.reverse(head);

printList(start);

}

}

}

// } Driver Code Ends

class CodingMaxima

{

//This method returns the head of the LL after deleting node with value d.

public static Node deleteNode(Node head,int d)

{

Node curr=head;

Node prev=null;

while(curr.data!=d){

prev=curr;

curr=curr.next;

}

prev.next=curr.next;

return head;

}

//This method returns the head node of the reversed Linked list.

public static Node reverse(Node head)

{

Node prev=null;

Node curr=head;

Node next=null;

while(curr.next!=head){

next=curr.next;

curr.next=prev;

prev=curr;

curr=next;

}

curr.next = prev;

head.next = curr;

return curr;

}

}