https://course.acciojob.com/idle?question=c8de2432-13d5-4a1e-885b-f1659a2ee7e3

Rotate List

Given the `head` node of a linked list, your task is to rotate the list to the right by `k` places.

Input Format

The first line of input contains a single integer `N`, the number of elements.

The second line of input contains `N` space-separated integers containing the nodes of the linked list.

The third line of input contains a single integer `k`.

Output Format

Your task is to return the head node after rotating the list to the right by `k` places.

Example 1

Input

```
6
12 35 1 10 34 1
2
```

Output

```
34 1 12 35 1 10
```

Explanation

The given linked list is `12->35->1->10->34->1`, **after roatating once we get** `1->12->35->10->34` **again roatating once we get** `34->1->12->35->1->10`.

Example 2

Input

```
4
1 2 3 4
5
```

Output

```
4 1 2 3
```

Explanation

5 rotation is similar to 1 rotation, so we get `4->1->2->3`.

Constraints

1 <= N <= 500

-100 <= value of node <= 100

0 <= k <= 2*(10^9)

Topic Tags

- **Linked lists**

My code

```
// n java
import java.util.*;
class Node {
    int data;
    Node next;
```

```
    Node(int d) {
        data = d;
        next = null;
    }
}

class Main {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int a = sc.nextInt();
        Node head = new Node(a);
        Node tail = head;

        for (int i=0; i<n-1; i++)
        {
            a = sc.nextInt();
            tail.next = new Node(a);
            tail = tail.next;
        }

        int k = sc.nextInt();

        Solution ob = new Solution();
        head = ob.rotate(head,k);
        printList(head);
```

```
    }

    public static void printList(Node n) {
        while (n != null) {
            System.out.print(n.data + " ");
            n = n.next;
        }
        System.out.println();
    }
}


class Solution{
    //Function to rotate a linked list.
    public Node rotate(Node head, int k) {
        // add code here
            if(k==0)
                    return head;
            Node p=head;
            int n=1;
            while(head.next!=null)
                    {
                            head=head.next;
                            n++;
                    }
            k=k%n;
            n=n-k;
            head.next=p;//here circuler
        Node ans=p;
```

```
        while(n>=1)
            {
                ans=ans.next;
               head=head.next;
                n--;
            }
        head.next=null;
        return ans;
    }
}
```