

## 牛客网-华为机试练习题 48

输入一个单向链表和一个节点的值，从单向链表中删除等于该值的节点，删除后如果链表中无节点则返回空指针。

链表结点定义如下：

```
struct ListNode
{
    int m_nKey;
    ListNode* m_pNext;
};
```

详细描述：

本题为考察链表的插入和删除知识。

链表的值不能重复

构造过程，例如

1 <- 2

3 <- 2

5 <- 1

4 <- 5

7 <- 2

最后的链表的顺序为 2 7 3 1 5 4

删除 结点 2

则结果为 7 3 1 5 4

### 输入描述:

- 1 输入链表结点个数
- 2 输入头结点的值
- 3 按照格式插入各个结点
- 4 输入要删除的结点的值

### 输出描述:

输出删除结点后的序列

示例1

### 输入

复制

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

## 输出

复制

```
2 1 5 4
```

## 解决代码：

```
import java.util.*;

/*****如果需要频繁地从列表的中间位置添加和移除元素，且只要顺序地访问列表元素时，LinkedList实现更好*****/
public class Main{
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        while(scanner.hasNext()){
            handle(scanner);
        }
        scanner.close();
    }
    public static void handle(Scanner scanner){
        String str = scanner.nextLine();
        LinkedList<Integer> linkedList = new LinkedList<Integer>();
        int loc = str.indexOf(" ");
        int num = Integer.parseInt(str.substring(0,loc));
        str = str.substring(loc+1);
        loc = str.indexOf(" ");
        linkedList.add(Integer.parseInt(str.substring(0,loc)));
        str = str.substring(loc+1);
        int start,end;
        end = -1;
        for(int i=0;i<num-1;i++){
            start = end+1;
            int loc1 = str.indexOf(" ",start);
            end = str.indexOf(" ",loc1+1);
            int node = Integer.parseInt(str.substring(start,loc1));
            int after = Integer.parseInt(str.substring(loc1+1,end));
            int location = linkedList.indexOf(after);
            linkedList.add(location+1,node);
        }
        int deleteData = Integer.parseInt(str.substring(end+1));
        linkedList.remove(new Integer(deleteData));
        Iterator<Integer> iterator = linkedList.iterator();
        while(iterator.hasNext()){
            System.out.print(iterator.next()+" ");
        }
        System.out.println();
    }
}
```