

牛客网-华为机试练习题 51

题目描述

输入一个单向链表，输出该链表中倒数第k个结点，链表的倒数第1个结点为链表的尾指针。

链表结点定义如下：

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

详细描述：

接口说明

原型：

```
ListNode* FindKthToTail(ListNode* pListHead, unsignedint k);
```

输入参数：

ListNode* pListHead 单向链表

unsigned int k 倒数第k个结点

输出参数（指针指向的内存区域保证有效）：

无

返回值：

正常返回倒数第k个结点指针，异常返回空指针

输入描述:

输入说明

- 1 输入链表结点个数
- 2 输入链表的值
- 3 输入k的值

输出描述:

输出一个整数

示例1

输入

```
8
1 2 3 4 5 6 7 8
4
```

输出

5

解决代码：

```
import java.util.Scanner;

/**
 * 输入一个单向链表，输出该链表中倒数第k个结点，链表的倒数第0个结点为链表的尾指针。
 */
class Node{
    int value ;
    Node next = null;
    Node(int a ){
        this.value = a;
    }
}

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        while(sc.hasNext()){
            int num = Integer.valueOf(sc.nextLine());
            String[] values = sc.nextLine().split(" ");
            int k = Integer.valueOf(sc.nextLine());
            //初始化链表
            Node head = createList(values);
            //找到倒数第K个链表节点
            if(k==0){
                System.out.println(0);
            }else{
                Node result = lastKValueList(head, k);
                System.out.println(result.value);
            }
        }
    }

    public static Node createList(String[] values) {
        if(values.length<=0){
            return null;
        }
        Node head = new Node(Integer.valueOf(values[0]));//创建头节点
        Node temp = head;
        for(int i=1;i<values.length;i++){
            temp.next = new Node(Integer.valueOf(values[i]));
            temp = temp.next;
        }
    }
}
```

```
    }  
    return head;  
}  
  
public static Node lastKValueList(Node head, int k) {  
    Node fast = head;  
    Node slow = head;  
  
    for(int i=0;i<k-1;i++){  
        fast = fast.next;  
    }  
  
    while(fast.next != null){  
        fast = fast.next;  
        slow = slow.next;  
    }  
    return slow;  
}  
}
```