面试题24：反转链表

题目描述

输入一个链表，反转链表后，输出新链表的表头。

解题思路

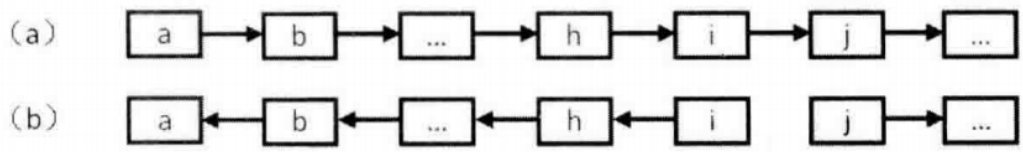

图 3.10　反转链表中节点的 m_pNext 指针导致链表出现断裂

注：（a）一个链表。（b）把 i 之前所有节点的 m_pNext 都指向前一个节点，导致链表在节点 i、j 之间断裂。

以这道题为例，我们至少应该想到以下几类测试用例对代码进行功能测试：

- 输入的链表头指针是 nullptr。
- 输入的链表只有一个节点。
- 输入的链表有多个节点。

如果我们确信代码能够通过这 3 类测试用例的测试，那么我们就有很大的把握能够通过这轮面试。

Java

```
1 /*
2 public class ListNode {
```

```
    int val;
    ListNode next = null;

    ListNode(int val) {
        this.val = val;
    }
}*/
public class Solution {
    public ListNode ReverseList(ListNode head) {
        //定义3个指针
        ListNode prev = null; //前一个节点
        ListNode node = null; //后一个节点
        ListNode current = head; //当前节点
        while(current != null){
            ListNode next = current.next;
            if(next == null){ //下一个节点为空
                node = current;
            }
            current.next = prev;
            prev = current;
            current = next;
        }
        return node;
    }
}
```

C++

```cpp
/*
struct ListNode {
    int val;
    struct ListNode *next;
    ListNode(int x) :
            val(x), next(NULL) {
    }
};*/
class Solution {
public:
    ListNode* ReverseList(ListNode* pHead) {
        //定义3个指针
        ListNode* prev = nullptr; //前一个节点
        ListNode* node = nullptr; //后一个节点
        ListNode* current = pHead; //当前节点
        while(current != nullptr){
            ListNode* next = current -> next;
            if(next == nullptr){ //下一个节点为空
                node = current;
            }
            current -> next = prev;
            prev = current;
            current = next;
        }
        return node;
    }
};
```