

解法一

解法二 不用栈，用递归

题目：输入一个链表的头节点，从尾到头反过来打印出每个节点的值。
链表节点定义如下：

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

解法一

用栈，先全都入栈；利用栈的先进后出的特点

void PrintListReverse(ListNode* head)

```
{
    if(!pHead)
        return;
    stack<ListNode*> s;
    while(head)
    {
        s.push(head);
        head=head->next;
    }
    while(!s.empty())
    {
        ListNode* t=s.top();
        printf("%d\t",t->val);
        s.pop();
    }
}
```

解法二 不用栈，用递归

递归访问当前节点的下一个节点，打印的时候就反过来了

void PrintListReverse(ListNode* head)

```
{
    if(head)
    {
        if(head->next)
            PrintListReverse(head->next);
    }
}
```

```
        printf("%d\t",head->val);
    }
}
```

```
class Solution {
public:
    vector<int> printListFromTailToHead(ListNode* head) {
        vector<int> res;
        return help(head,res);
    }
    vector<int> help(ListNode* head,vector<int> &res)
    {
        if(head)
        {
            if(head->next)
                help(head->next,res);
            res.push_back(head->val);
        }
        return res;
    }
};
```