

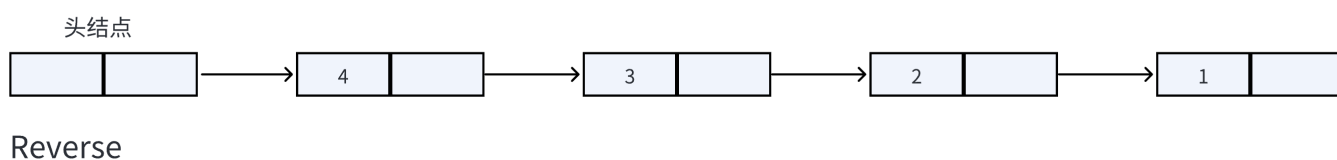
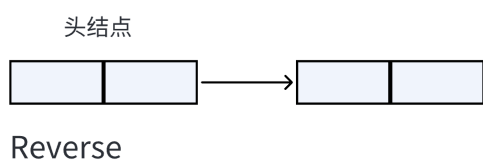
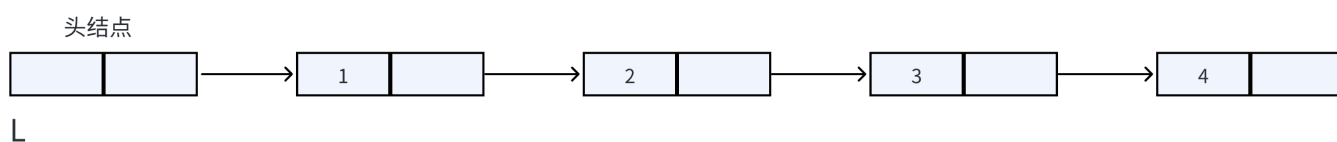
## 03、从尾到头输出结点值

24计算机考研成员一战成硕！



题目描述：

设 L 为带头结点的单链表，编写算法实现从尾到头反向输出每个结点的值



🕒 倒计时

## 1、知识点及难度



题解人：酒客

难度：中等

知识点：反转链表的方法

## 2、算法题

思路

（可以使用数组记录单链表的数据在反向输出，也可以使用头插法，这里是一种新的方法！！！！）

- 在遍历链表时，将当前节点的next 指针改为指向前一个节点。由于节点没有引用其前一个节点，因此必须事先存储其前一个节点。在更改引用之前，还需要存储后一个节点。最后返回新的头引用。
- 简单来说，就是用两个一前一后的节点，遍历整个链表，一直把后面（fast）那个节点指向前面（slow）那个节点，是不是就可以反转整个链表了呢？
- 但是有个细节，如果把后面（fast）那个节点的next指向前面（slow）那个节点，那么后面（fast）的这个节点的next会随之改变，无法指向原本的下一个节点，所以我们得提前新建一个节点指向fast节点的下一个节点

1.

基本实现-C++

```

1 void reverseList(ListNode* head) {
2     ListNode* slow = nullptr;
3     ListNode* fast = head;
4     //遍历整个链表，一直把后面 (fast) 那个节点指向前面 (slow) 那个节点
5     while( fast ) {
6         //用来保存fast节点的下一个节点
7         ListNode* postfast = fast->next;
8         fast->next = slow;
9         slow = fast;
10        fast = postfast;
11    }
12    ListNode* temp = head;
13    while( temp ) {
14        cout << temp->val << endl;
15    }
16 }

```

## 基本实现C

```

1 void reverseList(ListNode* head) {
2     ListNode* slow = nullptr;
3     ListNode* fast = head;
4     while( fast ) {
5         ListNode* postfast = fast->next;
6         fast->next = slow;
7         slow = fast;
8         fast = postfast;
9     }
10    ListNode* temp = head;
11    while( temp ) {
12        printf("%d",temp->val);
13    }
14 }

```

## 3、总结



### 总结栏

蓝蓝B站首页：[蓝蓝希望你上岸呀B站首页](#)

蓝蓝公众号：[算法训练营9分计划](#)

蓝蓝知识星球介绍：[📖 关于知识星球的权益](#)

### 如何在星球打卡记录：

- 计算机考研数据结构算法专项day[1/60]：
- 学习内容：最好能发出自己写的图片
- 遇到的问题：如果无就不用写了
- 小结：这部分一周写一次即可。