

```
void SListEraseAfter(SListNode* pos);
```

3.3 链表面试题

1. 删除链表中等于给定值 **val** 的所有节点。 [OJ链接](#)
2. 反转一个单链表。 [OJ链接](#)
3. 给定一个带头结点 head 的非空单链表，返回链表的中间结点。如果有两个中间结点，则返回第二个中间结点。 [OJ链接](#)
4. 输入一个链表，输出该链表中倒数第k个结点。 [OJ链接](#)
5. 将两个有序链表合并为一个新的有序链表并返回。新链表是通过拼接给定的两个链表的所有节点组成的。 [OJ链接](#)
6. 编写代码，以给定值x为基准将链表分割成两部分，所有小于x的结点排在大于或等于x的结点之前。 [OJ链接](#)
7. 链表的回文结构。 [OJ链接](#)
8. 输入两个链表，找出它们的第一个公共结点。 [OJ链接](#)
9. 给定一个链表，判断链表中是否有环。 [OJ链接](#)
10. 给定一个链表，返回链表开始入环的第一个节点。如果链表无环，则返回 NULL [OJ链接](#)
11. 给定一个链表，每个节点包含一个额外增加的随机指针，该指针可以指向链表中的任何节点或空节点。要求返回这个链表的深度拷贝。 [OJ链接](#)
12. 对链表进行插入排序。 [OJ链接](#)
13. 在一个排序的链表中，存在重复的结点，请删除该链表中重复的结点，重复的结点不保留，返回链表头指针。 [OJ链接](#)
14. 其他。ps：链表的题当前因为难度及知识面等等原因还不适合我们当前学习，以后大家自己下去以后 [Leetcode OJ链接](#) + [牛客 OJ链接](#)

```
// 2、带头+双向+循环链表增删查改实现
typedef int LTDataType;
typedef struct ListNode
{
    LTDataType _data;
    struct ListNode* _next;
    struct ListNode* _prev;
}ListNode;

// 创建返回链表的头结点.
ListNode* ListCreate();
// 双向链表销毁
void ListDestory(ListNode* plist);
// 双向链表打印
void ListPrint(ListNode* plist);
// 双向链表尾插
void ListPushBack(ListNode* plist, LTDataType x);
// 双向链表尾删
void ListPopBack(ListNode* plist);

// 双向链表头插
```