**难度: 简单**

给定单向链表的头指针和一个要删除的节点的值，定义一个函数删除该节点。

返回删除后的链表的头节点。

注意：此题对比原题有改动

示例 1:

输入: head = [4,5,1,9], val = 5

输出: [4,1,9]

解释: 给定你链表中值为 5 的第二个节点，那么在调用了你的函数之后，该链表应变为 4 -> 1 -> 9.

示例 2:

输入: head = [4,5,1,9], val = 1

输出: [4,5,9]

解释: 给定你链表中值为 1 的第三个节点，那么在调用了你的函数之后，该链表应变为 4 -> 5 -> 9.

说明：

题目保证链表中节点的值互不相同

若使用 C 或 C++ 语言，你不需要 free 或 delete 被删除的节点

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/shan-chu-lian-biao-de-jie-dian-lcof

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路:**

1). 先判断当前的节点是否是要删除的节点,如果是,直接返回当前节点的下一个节点

2). 如果1不成立,把头结点给到临时节点

3). 然后判断当前节点的节点不为空,且值不相同

4). 当循环结束时,节点的下一个节点就是我们要删除的节点

5). 所有此时用当前节点指向当前节点的下一个节点的下一个节点

6). 返回头结点

|  |
| --- |
| public ListNode deleteNode(ListNode head, int val) {  //判断当前的值是否是删除的节点  if (head.val == val) {  //如果是的话,直接放回下个节点开始  return head.next;  }  //当前节点  ListNode temp = head;  //循环 判断节点的下一个节点是否为空 并且 判断节点的下一个节点的值是否等于val  //如果不为空且断节点的下一个节点的值不等于val  while(temp.next != null && temp.next.val != val) {  //向下走一个节点  temp = temp.next;  }  //判断当前节点下一个节点是否为空  //并且此时我们的下一个节点的val一定是我们要找的值  if (temp.next != null) {  //直接让当前节点指向下一个节点的下一个节点  //即删除当前节点的下一个节点  temp.next = temp.next.next;  }  //返回头结点  return head;  } |