```
/*
请编写一个函数，使其可以删除某个链表中给定的（非末尾）节点，你将只被给定要求被删除的节点。

现有一个链表 -- head = [4,5,1,9]，它可以表示为:
4-->5-->1-->9
示例 1:
输入: head = [4,5,1,9], node = 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], node = 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:

    链表至少包含两个节点。
    链表中所有节点的值都是唯一的。
    给定的节点为非末尾节点并且一定是链表中的一个有效节点。
    不要从你的函数中返回任何结果。

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/delete-node-in-a-linked-list
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。
*/
```

分析:

- 从头开始遍历链表(需要两个指针,一个负责遍历,一个指向当前节点的前驱节点),当遍历到符合条件的节点时,先保存当前节点的指针到$temp$,然后让当前节点的前驱的$next$指向当前节点的$next$,再释放$temp$.(看错题目了)
- 阅读理解:输入节点其实是要删除的节点,由于没有传入前驱节点,那就只能从当前节点开始从后面往前面进行val值的覆盖,然后释放最后一个节点,并让倒数第二个节点的$next$指向NULL了.
- 方法二: 让传入的node的val被赋值为node->next->val,然后再删除node->next对应的节点即可.

方法一: C_滑动窗遍历

```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
void deleteNode(struct ListNode* node)
{
    struct ListNode*    front = node        ;
    struct ListNode*    cur   = node->next  ;

```

```
    while(cur->next)
    {
        front->val = cur->val   ;
        front      = cur        ;
        cur        = cur->next  ;
    }

    front->val = cur->val       ;
    free(front->next)           ;
    front->next = NULL          ;

    return ;
}

/*
执行结果:
通过
显示详情
执行用时 :8 ms, 在所有 C 提交中击败了87.65% 的用户
内存消耗 :7.7 MB, 在所有 C 提交中击败了5.09%的用户
*/
```

方法二:C_删除下一个节点

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
void deleteNode(struct ListNode* node)
{
    struct ListNode*    temp = NULL ;

    temp = node->next;
    node->val = temp->val;
    node->next = temp->next;
    free(temp);
    return ;
}
/*
执行结果:
通过
显示详情
执行用时 :4 ms, 在所有 C 提交中击败了99.09% 的用户
内存消耗 :7.7 MB, 在所有 C 提交中击败了5.09%的用户
*/
```

AlimyBreak
2019.08.22