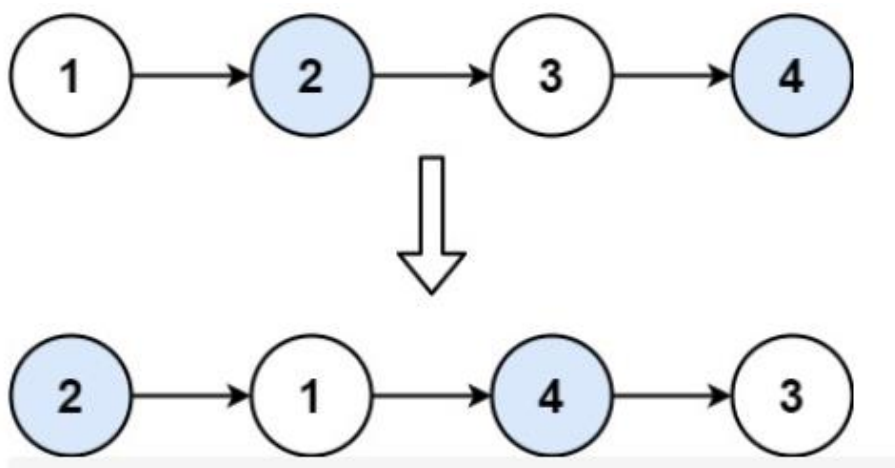


## 两两交换链表中的节点

给定一个链表，两两交换其中相邻的节点，并返回交换后的链表。

你不能只是单纯的改变节点内部的值，而是需要实际的进行节点交换。

示例 1:



输入: head = [1,2,3,4]

输出: [2,1,4,3]

示例 2:

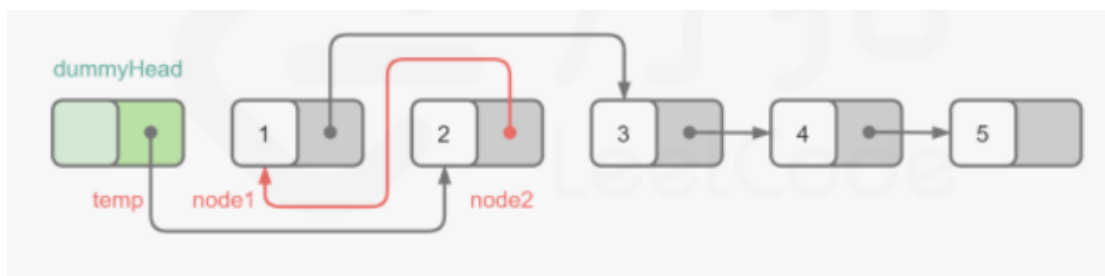
输入: head = []

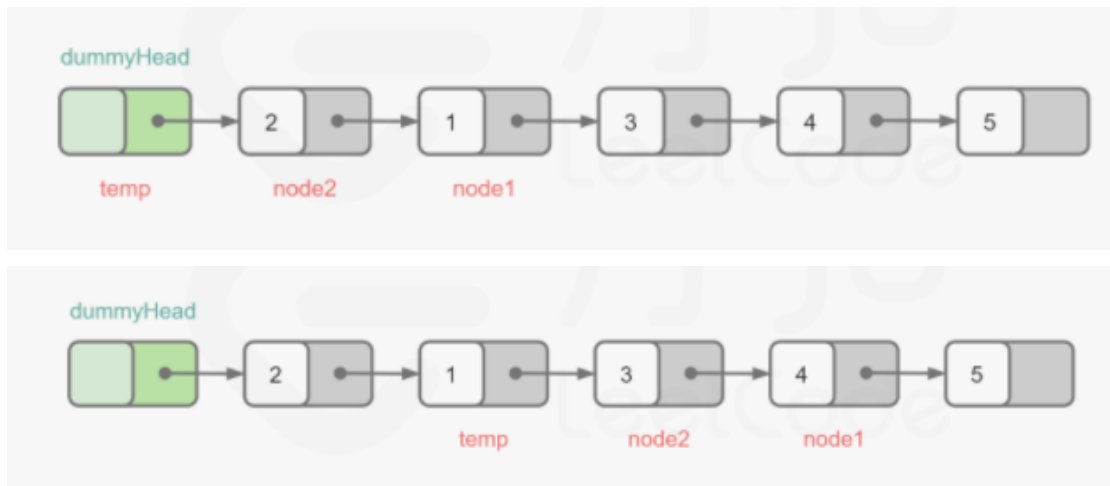
输出: []

示例 3:

输入: head = [1]

输出: [1]





```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
class Solution {
public:
    ListNode* swapPairs(ListNode* head) {
        ListNode* newlist=new ListNode;
        newlist->next=head;
        ListNode* cur=newlist;
        while(cur->next!=nullptr&&cur->next->next!=nullptr)
        {
            ListNode* node1=cur->next;
            ListNode* node2=cur->next->next;
            //改变指向
            cur->next=node2;
            node1->next=node2->next;
            node2->next=node1;
            //更新 cur
            cur=node1;
        }
        return newlist->next;
    }
};
```