

K 个一组翻转链表

给你一个链表，每 k 个节点一组进行翻转，请你返回翻转后的链表。

k 是一个正整数，它的值小于或等于链表的长度。

如果节点总数不是 k 的整数倍，那么请将最后剩余的节点保持原有顺序。

进阶：

- 你可以设计一个只使用常数额外空间的算法来解决此问题吗？
- 你不能只是单纯的改变节点内部的值，而是需要实际进行节点交换。

示例 1：

输入：head = [1,2,3,4,5], k = 2

输出：[2,1,4,3,5]

示例 2：

输入：head = [1,2,3,4,5], k = 3

输出：[3,2,1,4,5]

示例 3：

输入：head = [1,2,3,4,5], k = 1

输出：[1,2,3,4,5]

```
/**
 * 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* reverseKGroup(ListNode* head, int k) {
    ListNode sentinel(0, head), *trunk = &sentinel;
    while (true) {
        bool finished = false;
        ListNode *p = trunk;
        for (int i = 0; i < k; ++i) {
            if (p->next)
                p = p->next;
            else {
                finished = true;
                break;
            }
        }
        if (finished)
            break;
        head = p->next;
        p = trunk->next;
        while (p->next != head) {
            ListNode *tmp = trunk->next;
            trunk->next = p->next;
            p->next = trunk->next->next;
            trunk->next->next = tmp;
        }
        trunk = p;
    }
    return sentinel.next;
}
};

```