難度 2.6 / 5

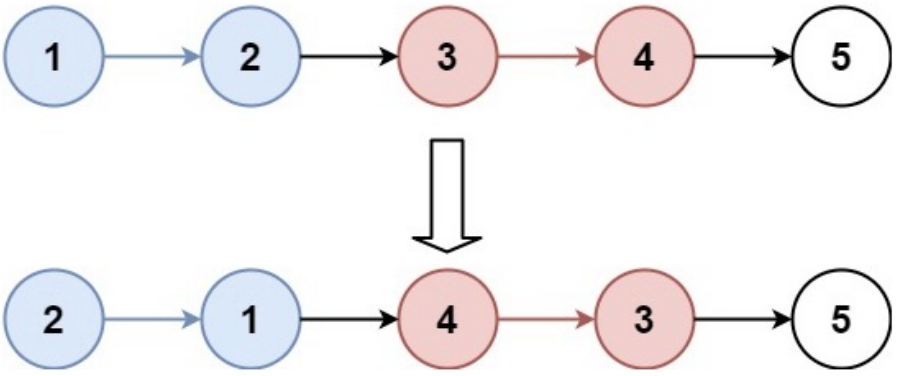
題目敘述

Given the `head` of a linked list, reverse the nodes of the list `k` at a time, and return *the modified list*.

`k` is a positive integer and is less than or equal to the length of the linked list. If the number of nodes is not a multiple of `k` then left-out nodes, in the end, should remain as it is.

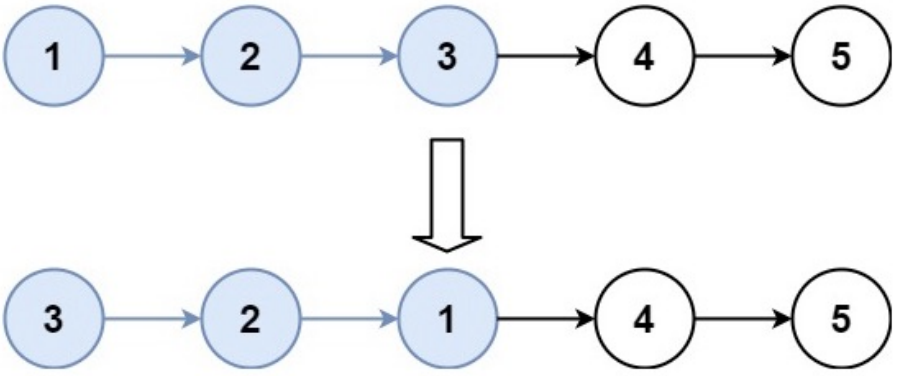
You may not alter the values in the list's nodes, only nodes themselves may be changed.

Example 1:



> **Input:** head = [1,2,3,4,5], k = 2
> **Output:** [2,1,4,3,5]

Example 2:



> **Input:** head = [1,2,3,4,5], k = 3
> **Output:** [3,2,1,4,5]

```
struct Node {
    int val;
    Node *next;
    Node() : val(0), next(nullptr) {}
    Node(int x) : val(x), next(nullptr) {}
    Node(int x, Node *next) : val(x), next(next) {}
};

class Solution {
public:
    Node* solve(Node* head, int k) {
        if (!head) return nullptr;
```

```
        Node *ok = head;
        for (int i = 0; i < k; ++i) {
            if (!ok) return head;
            ok = ok->next;
        }

        Node *nxt = nullptr, *prev = nullptr;
        Node *now = head;

        int cnt = 0;
        while (now and cnt < k) {
            nxt = now->next;
            now->next = prev;
            prev = now;
            now = nxt;
            ++cnt;
        }

        if (nxt) head->next = solve(nxt, k);
        return prev;
    }
};
```