class Solution {

public:

ListNode\* rotateRight(ListNode\* head, int k) {

if (head == nullptr || k == 0) {

return head;

}

int length = 1; // Initialize length to 1 (at least one node)

ListNode\* tail = head;

while (tail->next != nullptr) {

tail = tail->next;

length++;

}

k = k % length; // Effective rotations (handles k > length)

if (k == 0) {

return head; // No rotation needed

}

int rotationsNeeded = length - k;

ListNode\* newTail = head;

for (int i = 1; i < rotationsNeeded; i++) {

newTail = newTail->next;

}

ListNode\* newHead = newTail->next;

newTail->next = nullptr; // Break the list at the rotation point

tail->next = head; // Connect the original tail to the original head

return newHead;

}

};