

```

// Function to rotate a linked list.
// rotate the linked list counter-clockwise by k nodes, where k is a given
// positive integer smaller than or equal to length of the linked
public Node rotateLinkedList(Node head, int k) {
    // no need to rotate.
    if (k == 0) {
        return head;
    }
    int cnt = 0;
    Node curr = head;
    Node prev = null;    // previous pointer of current node
    Node newHead = null; // reference of new head pointer.

    while (curr != null) {
        // node which will be new head pointer after k rotation.
        if (cnt == k) {
            newHead = curr;    // mark the current node as new head pointer.
            prev.next = null; // disjoint(break) the linked list at new head pointer.
        }
        cnt++;
        prev = curr;
        curr = curr.next;
    }
    // k == length of linked list(no need to rotate)
    if (newHead == null) {
        return head;
    }
    // join the previous head pointer with the last node.
    prev.next = head;
    return newHead;
}

```