Rotate Linked List K places LeetCode

Implement a program to rotate a singly linked list by 'k' positions in a clockwise direction. Given a linked list and a positive integer 'k', perform the rotation and print the resulting linked list.

Example:

Input:

Linked List: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8

k = 6

Output:

Linked List after rotating 6 places: 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 1 -> 2

Approach 1: Function to rotate the linked list by 'k' positions

1. Calculate the length of the linked list.
2. Normalize 'k' to a value less than the length. If 'k' is greater than or equal to the length, calculate 'k' modulo length.
3. If 'k' becomes zero or negative, no rotation is needed. Return the original head.
4. Traverse the linked list to the position (length - k) to find the new head after rotation.
5. Update the pointers to perform the rotation and return the new head.
6. **Time Complexity:**
    - Calculating the length: O(n)
    - Traversing to the new head: O(n)
    - **Total time complexity: O(n)**
7. **Space Complexity:**
    - Additional space used for pointers and variables: O(1)
    - **The overall space complexity is O(1).**