Reverse Nodes in k-Group.

This problem is an extension of reversing a linked list, but done in chunks of size k; the approach is iterative group reversal.

Key Idea:
- Traverse the list in chunks of k.
- For each chunk:
  - Check if there are at least k nodes left; if not, leave them as is.
  - Reverse the k nodes.
  - Connect the reversed chunk to the previous part of the list.
- Use a dummy node to simplify edge cases (like reversing the first group).

Steps:
1. Count nodes in the list to know when to stop.
2. Create a dummy node pointing to head.
3. Use prev pointer to mark the node before the current group.
4. While there are at least k nodes left:
   - Mark start = prev → next (first node of group) and end = start → next.
   - Reverse nodes between start and end for k-1 iterations:
     - Move end after prev (standard in-place reversal trick).
5. Return dummy → next.

Time Complexity:
- O(n) - each node is visited and reversed once
- O(1) extra space.