

1. [Delete Node in a Linked List](#)
2. [Remove Linked List Elements](#)
3. [Merge Two Sorted List](#)
4. [Linked List Cycle](#)
5. [Remove the Nth node from the linked list](#)
6. Given a singly linked list of size N. The task is to left-shift the linked list by k nodes, where k is a given positive integer smaller than or equal to the length of the linked list.  
Input: N = 5  
value[] = {2, 4, 7, 8, 9}  
k = 3  
Output: 8 9 2 4 7  
Explanation: Rotate 1: 4 -> 7 -> 8 -> 9 -> 2  
Rotate 2: 7 -> 8 -> 9 -> 2 -> 4  
Rotate 3: 8 -> 9 -> 2 -> 4 -> 7

7. Given the head of a linked list, we repeatedly delete consecutive sequences of nodes that sum to 0 until there are no such sequences.

After doing so, return the head of the final linked list. You may return any such answer.

(Note that in the examples below, all sequences are serializations of `ListNode` objects.)

Input: head = [1,2,-3,3,1]

Output: [3,1]

Note: The answer [1,2,1] would also be accepted.

8. Given the head of a singly linked list, group all the nodes with odd indices together followed by the nodes with even indices, and return *the reordered list*.

The **first** node is considered **odd**, and the **second** node is **even**, and so on.

Note that the relative order inside both the even and odd groups should remain as it was in the input.

You must solve the problem in  $O(1)$  extra space complexity and  $O(n)$  time complexity.

Input: head = [1,2,3,4,5]

Output: [1,3,5,2,4]

