Always take temporary node in iterative and try to think recursive solution

1. Remove duplicates from Linked list
    a. 2 ways iterative and recursive
    b. https://leetcode.com/problems/remove-duplicates-from-sorted-list/submissions/
    c. https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii/
2. Reverse Linked List
    a. https://leetcode.com/problems/reverse-linked-list-ii/solution/

```
ListNode* reverseList(ListNode* head) {
        ListNode *cur = head, *prev = NULL, *next;
        if(!head) return head;
        while(cur){
            next = cur->next;
            cur->next = prev;
            prev = cur;
            cur = next;
        }
        return prev;
    }
```

    b. https://leetcode.com/submissions/detail/385090306/ (recursive)
3. Merge two sorted list
    a. https://www.interviewbit.com/problems/merge-two-sorted-lists/
4. Check palindrome in O(1) space
    a. https://leetcode.com/problems/palindrome-linked-list/submissions/
5. Add two numbers without reversing input
    a. https://leetcode.com/problems/add-two-numbers-ii/submissions/
    b. Add both no into stack
    c. Now get two digits from two stack and front add to list
6. Detect cycle in linked list
    a. Fast slow pointer concept
    b. Once they intersect put slow at head and inc both 1 step again when they intersect that is the point where cycle is generated.
    c. https://leetcode.com/problems/linked-list-cycle-ii/solution/
    d. Do see complexity analysis https://leetcode.com/problems/linked-list-cycle/solution/
7. Split LL into k parts
    a. https://leetcode.com/problems/split-linked-list-in-parts/submissions/
8. Copy list with random pointers
    a. https://leetcode.com/problems/copy-list-with-random-pointer/
        i. Traverse a list and create a clone list and mapping of address
        ii. Second traversal and update random pointers
    b. https://leetcode.com/problems/copy-list-with-random-pointer/discuss/43491/A-solution-with-constant-space-complexity-O(1)-and-linear-time-complexity-O(N)
        - Iterate the original list and duplicate each node. The duplicate of each node follows its original immediately.
        - Iterate the new list and assign the random pointer for each duplicated node.
        - Restore the original list and extract the duplicatenodes.
9. https://leetcode.com/problems/flatten-a-multilevel-doubly-linked-list/
10. Sort linked list : https://leetcode.com/problems/sort-list
    a. Costat space merge sort
    b. Recursive merge sort
11. https://www.geeksforgeeks.org/sort-linked-list-0s-1s-2s-changing-links/