

## Smart Coding & Interview Series

### Top-20 Basic Program (Linked List Problems)

---

First, understand the solution building strategies and coding for the problems in LIVE/VIDEO session and then you apply those strategies discussed in LIVE/VIDEO session to solve the following problems. Use your favourite language(C/C++/Java/C#/Python/Scala) for coding.

**1) Remove Duplicates from Sorted List:** Given a sorted linked list, delete all duplicates such that each element appear only once.

**Example:**

Input: 1->1->2

Output: 1->2

**Source:** <https://leetcode.com/problems/remove-duplicates-from-sorted-list/description/>

**2) Merge Two Sorted Lists:** Merge two sorted linked lists and return it as a new **sorted** list. The new list should be made by splicing together the nodes of the first two lists.

**Example:**

Input: 1->2->4, 1->3->4

Output: 1->1->2->3->4->4

**Source:** <https://leetcode.com/problems/merge-two-sorted-lists/description/>

**3) Reverse Nodes in k-Group:** Given a linked list, reverse the nodes of a linked list  $k$  at a time and return its modified list.  $k$  is a positive integer and is less than or equal to the length of the linked list. If the number of nodes is not a multiple of  $k$  then left-out nodes in the end should remain as it is.

**Example:**

Given this linked list: 1->2->3->4->5

For  $k = 2$ , you should return: 2->1->4->3->5

For  $k = 3$ , you should return: 3->2->1->4->5

**Source:** <https://leetcode.com/problems/reverse-nodes-in-k-group/description/>

**4) Linked List Cycle II:** Given a linked list, return the node where the cycle begins. If there is no cycle, return null. To represent a cycle in the given linked list, we use an integer  $pos$  which represents the position (0-indexed) in the linked list where tail connects to. If  $pos$  is -1, then there is no cycle in the linked list. Do not modify the linked list.

**Example:**

Input: head = [3,2,0,-4], pos = 1

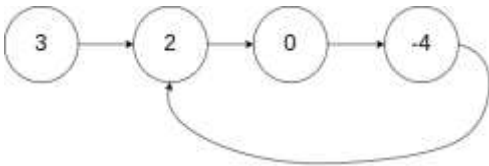
Output: tail connects to node index 1

Explanation: There is a cycle in the linked list, where tail connects to the second node.

## Smart Coding & Interview Series

### Top-20 Basic Program (Linked List Problems)

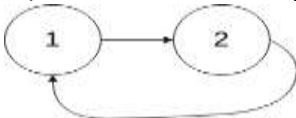
---



**Example:** Input: head = [1,2], pos = 0

Output: tail connects to node index 0

Explanation: There is a cycle in the linked list, where tail connects to the first node.

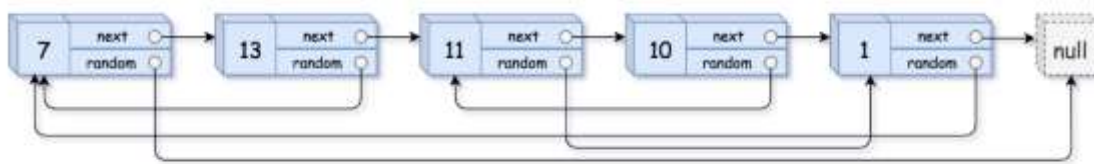


**Source :** <https://leetcode.com/problems/linked-list-cycle-ii/description/>

**5) Copy List with Random Pointer:** A linked list is given such that each node contains an additional random pointer which could point to any node in the list or null. Return a deep copy of the list. The Linked List is represented in the input/output as a list of  $n$  nodes. Each node is represented as a pair of [val, random\_index] where:

- val: an integer representing Node.val
- random\_index: the index of the node (range from 0 to  $n-1$ ) where random pointer points to, or null if it does not point to any node.

**Example:**



Input: head = [[7,null],[13,0],[11,4],[10,2],[1,0]]

Output: [[7,null],[13,0],[11,4],[10,2],[1,0]]

**Source:** <https://leetcode.com/problems/copy-list-with-random-pointer/description/>