UCS301 Data Structures
Lab Assignment 5: Single Linked List
(Week 5)

1. Develop a menu driven program for the following operations on a Singly Linked List.
   (a) Insertion at the beginning.
   (b) Insertion at the end.
   (c) Insertion in between (before or after a node having a specific value, say 'Insert a new Node 35 before/after the Node 30').
   (d) Deletion from the beginning.
   (e) Deletion from the end.
   (f) Deletion of a specific node, say 'Delete Node 60').
   (g) Search for a node and display its position from head.
   (h) Display all the node values.

2. Write a program to count the number of occurrences of a given key in a singly linked list and then delete all the occurrences.

   Input: Linked List : 1->2->1->2->1->3->1 , key: 1
   Output: Count: 4 , Updated Linked List: 2->2->3.

3. Write a program to find the middle of a linked list.

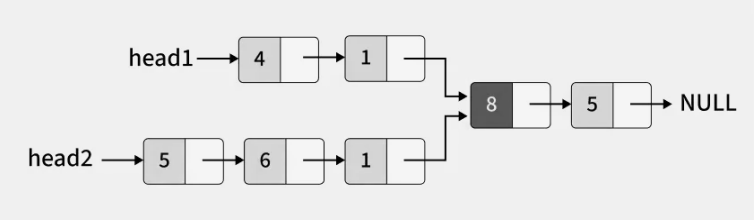
   Input: 1->2->3->4->5
   Output: 3

4. Write a program to reverse a linked list.

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

Additional Questions: Asked multiple times in FAANG+ companies

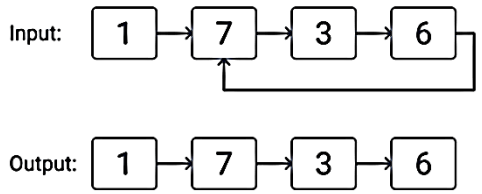
1. Find the intersection node of two singly linked lists that merge into a Y-shaped structure. The lists may vary in length and have distinct nodes at the beginning, but from the point of intersection onward, they share the same sequence of nodes. The task is to identify the first common node where the two lists converge. If the two linked lists have no intersection at all, return null.

Input: listA = [4,1,8,5], listB = [5,6,1,8,5]  **Output:** Intersected at 8 **Note:** Note that the intersected node's value is not 1 because they point to two different locations in memory, while the nodes with value 8 in listA and listB point to the same location in memory.	https://workat.tech/problem-solving/approach/itll/intersection-two-linked-lists

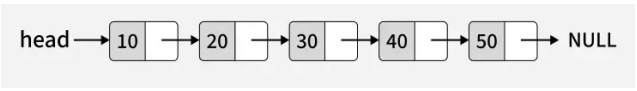
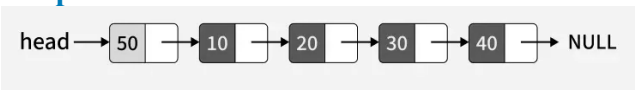
2. Given a linked list and a positive number K, reverse the nodes in groups of K. All the remaining nodes after multiples of k should be left as it is.

Example 1: **Input:** Linked list: 1→2→3→4→5→6→7→8→9 ,K: 3 **Output:** Result: 3→2→1→6→5→4→9→8→7 **Example 2** **Input:** Linked list: 1→2→3→4→5→6→7→8 , K=3 **Output:** Result: 3→2→1→6→5→4→7→8	https://workat.tech/problem-solving/approach/rllkg/reverse-linked-list-k-group

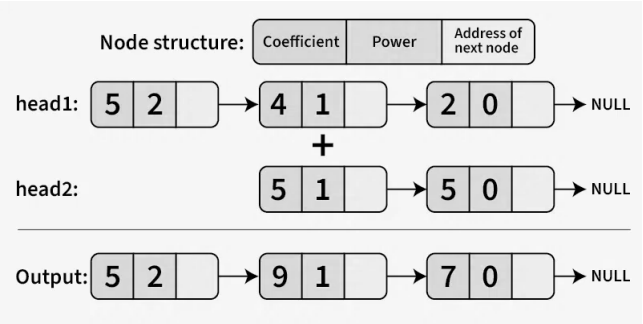
3. Given a linked list, remove the loop if it exists.

 	https://workat.tech/problem-solving/approach/rlll/remove-loop-linked-list

4. Given a linked list, and an integer k, rotate the list to the left by k positions and return the updated head.

Input: k = 4  **Output:**  	https://www.geeksforgeeks.org/dsa/rotate-a-linked-list/

5. Given two polynomial numbers represented by two linked lists. The task is to add these lists (meaning the coefficients with the same variable powers will be added).

Input: list1: [[5, 2], [4, 1], [2, 0]] list2: [[5, 1], [5, 0]]  **Output:** *[[5, 2], [9, 1], [7, 0]]*	https://www.geeksforgeeks.org/dsa/adding-two-polynomials-using-linked-list/