# Linked List Practice Problems

| 1. Write a function which returns a singly linked list with the difference of the previous and next node inserted in between the nodes.  Input: 1 → 5 → 7 → 8 → 10 Output: 1 → 4 → 5 → 2 → 7 → 1 → 8 → 2 → 10 2. Write a function that swaps every two adjacent nodes of a Linked List. Assume that the LL will have an even number of nodes.  Input: 1 → 2 → 3 → 4 → 5 → 6 Output: 2 → 1 → 4 → 3 → 6 → 5 3. You are given two sorted linked lists in ascending order. Your function should merge the two linked lists so that the new linked list also becomes sorted.  Input:  L1: 1 → 2 → 6 → 10 L2: 2 → 3 → 4 → 7 → 12 Output: 1 → 2 → 2 → 3 → 4 → 6 → 7 → 10 → 2 4. Write a function which will arrange all values smaller than 10 at the beginning of the linked list Input: 1→3→12→6→14→8 Output: 1→3→6→8→12→14 5. Reverse a linked list Input: 1→3→5 → 7 Output: 7 → 5 → 3 → 1   **Solutions of the problems (First try to solve by yourself):** [LL Practice Problems.ipynb](https://colab.research.google.com/drive/1VwvwUluo9Byzj0yEJexj4kaA3rVR-Si3?usp=sharing) |
| --- |