DSA Tutorial - 2

Doubly and Circular Linked Lists

Problems for practice:

1. Doubly Linked List Insertion
   https://practice.geeksforgeeks.org/problems/insert-a-node-in-doubly-linked-list/1
2. Doubly Linked List deletion
   https://practice.geeksforgeeks.org/problems/delete-node-in-doubly-linked-list/1
3. Inserting a Node Into a Sorted Doubly Linked List
   https://www.hackerrank.com/challenges/insert-a-node-into-a-sorted-doubly-linked-list/problem
4. Reverse a Doubly Linked List
   https://www.hackerrank.com/challenges/reverse-a-doubly-linked-list/problem
5. Sorted insert in Circular Linked List
   https://practice.geeksforgeeks.org/problems/sorted-insert-for-circular-linked-list/1
6. Rotate Doubly Linked List
   https://practice.geeksforgeeks.org/problems/rotate-doubly-linked-list-by-p-nodes/1

   Try implementing these, you can't submit for these, so try a few cases locally, and maybe try to prove that your solution is indeed correct.
7. Delete all nodes smaller than given value in a Doubly Linked List
   https://www.geeksforgeeks.org/delete-all-the-nodes-from-a-doubly-linked-list-that-are-smaller-than-a-given-value/amp/
8. Delete all prime nodes from a Doubly Linked List
   https://www.geeksforgeeks.org/delete-all-prime-nodes-from-a-doubly-linked-list/amp/
9. Delete all even nodes of a Circular Linked List
   https://www.geeksforgeeks.org/delete-all-the-even-nodes-of-a-circular-linked-list/amp/
10. Exchange first and last node of Circular Linked List
    https://www.geeksforgeeks.org/exchange-first-last-node-circular-linked-list/amp/

Codes used in Tutorial:

1. Doubly Linked List : https://ideone.com/bUmVQd
2. Circular Linked List : https://ideone.com/3noaD0

Few tips for practice:

1. Avoid using pre-written code (if possible), that might be provided to you at platforms like HackerRank. Write code from scratch, like you have to do during Lab.
2. Avoid using given code as much as possible, and write your own code. If yours doesn't work, then try to look at the code given / google to see what/how else you could have done it.
3. Try to think about everything you're doing. Visualize the arrows (for Linked Lists), and not memorize the code.
4. Do try to implement Circular Singly Linked list (might need tail also). Look at implementations online. Although, what we covered today is enough, instead of having convenient access to tail (as head->prev), now you will need to explicitly maintain tail. Also, beware of insertion or deletion of last node, now you will also need to rectify tail.