CS 252 Data Structures Lab Cycle Problems for A.Y. 2022-2023

1. Find the duplicate in an array of N+1 integers
2. Given an array that contains only 1s and 0s return the count of maximum consecutive ones in the array.
3. Perform the following operations on a singly linked list:
    a. Insert a node at the head of a linked list.
    b. Insert a node at the tail of a linked list.
    c. Insert node at a specific position of a linked list.
    d. Insert node at appropriate position in a sorted linked list.
    e. Delete node from a linked list based on position.
    f. Delete node from a linked list based on value.
    g. Print the elements of a linked list
    h. Delete duplicate nodes from a linked list.
    i. Remove N-th node from the end of a Linked List
    j. Merge two sorted Linked Lists
    k. Find middle element in a Linked List
    l. Sort the elements in a linked list
    m. Detect a cycle in a linked list
4. Perform the following operations on a doubly linked list:
    a. Insert a node at the head of a linked list.
    b. Insert a node at the tail of a linked list.
    c. Insert node at a specific position of a linked list.
    d. Delete node from a linked list based on position.
    e. Delete node from a linked list based on value.
    f. Print the elements of a linked list
    g. Reverse a Linked List
5. Perform the following operations on a circular linked list:
    a. Insert a node at the beginning of a linked list.
    b. Insert a node at the end of a linked list.
    c. Insert node at a specific position of a linked list.
    d. Delete node from a linked list based on position.
    e. Delete node from a linked list based on value.
    f. Print the elements of a linked list
6. Find the sum/ product of two polynomials represented by using linked lists.
7. Implement separate chaining technique.
8. Check for Balanced Parentheses in the given infix expression using stack.
9. Convert the given infix expression to postfix form and evaluate it.
10. Implement dynamically linked Stack / Queue.
11. Implement Circular Queue.
12. Implement BFS using a Queue.
13. Perform following operations on a BST:
    a. Construct BST from given keys

b. Delete a given key from a BST.

c. Find the inorder predecessor/successor of a given Key in BST.

d. Implement following traversals: Preorder, Inorder, Postorder

14. Implement heap sort on a given list of keys
15. Represent a directed graph using adjacency list/matrix and determine the in-degree and out-degree of all nodes.