1. **Create Binary tree and perform following operations:**
   a. **Create**
   b. **Insert**
   c. **Display (using all 3 traversals with recursion)**

2. **Create Binary tree and perform following operations:**
   a. **Create**
   b. **Display In-order traversal**
   c. **Depth of a tree**

3. **Create Binary tree and perform following operations:**
   a. **Create**
   b. **Display In-order traversal**
   c. **Create a copy of a tree**

4. **Create Binary tree and perform following operations:**
   d. **Create**
   e. **Display In-order traversal**
   f. **Display leaf-nodes**

5. **Construct and expression tree from postfix/prefix expression and perform recursive and non-recursive In-order, pre-order and post-order traversals.**
   Input: postfix expression
   a. **Create tree**
   b. **All traversals**

6. **Construct and expression tree from postfix/prefix expression and perform recursive and non-recursive In-order, pre-order and post-order traversals.**
   Input: prefix expression
   a. **Create tree**
   b. **All traversals**

7. **Implement binary search tree and perform following operations:**
   a. **Insert**
   b. **Display In-order traversal**
   c. **Search**

8. **Implement binary search tree and perform following operations:**
   a. **Insert**
   b. **Display In-order traversal**
   c. **Mirror image**

9. **Implement binary search tree and perform following operations:**
    a. **Insert**
    b. **Display In-order traversal**
    c. **Display level wise**

10. **Implement binary search tree and perform following operations:**
    a. **Insert**
    b. **Display In-order traversal**
    c. **Delete (Case 1: Node to be deleted is the leaf)**

11. **Implement binary search tree and perform following operations:**
    a. **Insert**
    b. **Display In-order traversal**
    c. **Delete (Case 2:Node to be deleted has only one child)**

12. **Implement binary search tree and perform following operations:**
    a. **Insert**
    b. **Display In-order traversal**
    c. **Delete (Case 3: Node to be deleted has two children)**

13. **Represent any real-world graph using adjacency matrix:**
    a. **Create a graph**
    b. **Display**

Note:

Problem statement no. 13 is only for CORE division students.