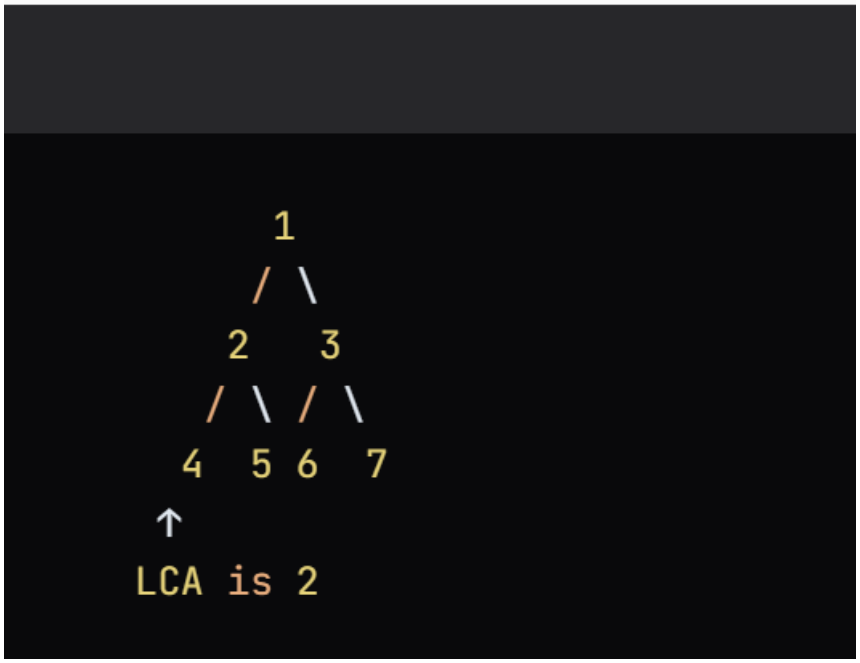
Practical 13

Aim : Write a program for Lowest Common Ancestors

. What is the Lowest Common Ancestor (LCA)?

The **Lowest Common Ancestor** of two nodes in a binary tree is the deepest node (the one that is farthest down) that is an ancestor of both nodes. An ancestor is any node that is on the path from the root of the tree to that node.



Algorithm :

Program :-

```
// Node class representing a node in the tree
class Node {
    int data;
    Node left, right;

    Node(int data) {
        this.data = data;
        this.left = null;
        this.right = null;
    }
}

// Class to find the Lowest Common Ancestor (LCA) of two nodes
public class LowestCommonAncestor {
```

```
    // Method to find the Lowest Common Ancestor (LCA) of two nodes
    public static Node findLCA(Node root, int n1, int n2) {
        // Base case: Empty tree or root matches one of the nodes
        if (root == null || root.data == n1 || root.data == n2) {
            return root;
        }

        // Recursively search for n1 and n2 in left and right subtrees
        Node leftLCA = findLCA(root.left, n1, n2);
        Node rightLCA = findLCA(root.right, n1, n2);

        // If both left and right subtrees found a node, then root is LCA
        if (leftLCA != null && rightLCA != null) {
            return root;
        }

        // Return the only node found (if one is in left and the other in right)
        return (leftLCA != null) ? leftLCA : rightLCA;
```

```
    }

    // Main method to create a sample tree and find LCA
    public static void main(String[] args) {
        // Create a sample binary tree
        Node root = new Node(1);
        root.left = new Node(2);
        root.right = new Node(3);
        root.left.left = new Node(4);
        root.left.right = new Node(5);
        root.right.left = new Node(6);
        root.right.right = new Node(7);

        // Find LCA of nodes 4 and 7
        int n1 = 4, n2 = 7;
        Node lca = findLCA(root, n1, n2);
        if (lca != null) {
            System.out.println("LCA of " + n1 + " and " + n2 + " is: "
+ lca.data);
        } else {
            System.out.println("LCA not found.");
        }
```

```
    }
}
```

```
LCA of 4 and 7 is: 1
```