Practice Questions

Solution 1 :-

```
class Solution {
   // Function to return a list containing the inorder traversal of the tree.
   private void traverse(Node root, ArrayList<Integer> al)
   {
      if(root==null) return;
      traverse(root.left, al);
      al.add(root.data);
      traverse(root.right,al);

   }
   ArrayList<Integer> inOrder(Node root)
   {
      // Code
      ArrayList<Integer> al = new ArrayList<>();
      traverse(root,al);
      return al;
   }
}
```

Solution 2 :-

```
class Solution

{

   // Return True if the given trees are isomotphic. Else return False.

   boolean isIsomorphic(Node root1, Node root2)

   {

       // code here.
```

```
         if(root1==null && root2==null)
            return true;

        if(root1==null || root2==null ){
            return false;
        }

        if(root1.data!=root2.data)
            return false;

        boolean option1= isIsomorphic(root1.left, root2.left) || isIsomorphic(root1.left, root2.right);
        boolean option2= isIsomorphic(root1.right, root2.right) || isIsomorphic(root1.right, root2.left);

        return (option1 && option2);
    }

}
```