Daily Work Practice Date: 4/8/2023

Problem-1:

```
class Solution {
     public boolean isSameTree(TreeNode p, TreeNode q) { //isSameTree
function is created with TreeNode p and TreeNode q parameter

    if(p == null && q == null) return true; //if both node does'nt have
any node then also it will return true;

    if(p != null && q == null || p == null && q != null ) return
false;//if any one node is not null then it will return false

    if(p.val != q.val) return false;   //if there is no same binary
tree then return false

    boolean leftTreeMatch = isSameTree(p.left, q.left);    //call the
left tree's node

    boolean rightTreeMatch = isSameTree(p.right, q.right);//call the
right tree's node to check it's same or not

         return leftTreeMatch && rightTreeMatch;//if any one mismatch
then return false otherwise return true
     }
}
```

Problem-2

```
class Solution {

   public: int[] twoSum(int[] nums , int target){//two parameter passing

       for(int i=0; i<nums.length; i++){         //first loops iteration

           for(int j=i+1; j<nums.length; j++){   //second loops iteration

            if(nums[i] +nums[j]==target){          //If this condition is
                                                   satisfied then we have to
                                                   return array
                  int a[] ={i,j};                 //creating array

                  return a;                       //return value
               }
           }
       }
           return null;
   }
}
```