4. Find All Duplicates in an Array

Given an integer array nums of length n where all the integers of nums are in the range [1, n] and each integer appears once or twice, return an array of all the integers that appears twice.

You must write an algorithm that runs in O(n) time and uses only constant extra space.

```
//Time complexity: O(n)

//Space complexity: O(n)

class Solution {

public:

    vector<int> findDuplicates(vector<int>& nums) {

        vector <int> ans;

        int n=nums.size();

        map<int,int>mp;

        for(auto x:nums)

        {

            mp[x]++;

        }

        for(auto x: nums)

        {

        if(mp[x]>1)

        {

            ans.push_back(x);

            mp[x]--;

        }

        }
        return ans;
    }
};
```

5. Find the Distance Value Between Two Arrays

Given two integer arrays arr1 and arr2, and the integer d, return the distance value between the two arrays.

The distance value is defined as the number of elements arr1[i] such that there is not any element arr2[j] where |arr1[i]-arr2[j]| <= d.

```
class Solution {

public:

    int findTheDistanceValue(vector<int>& arr1, vector<int>& arr2, int d) {

        //Brute Force Approach Time O(arr1*arr2)

        int sum=0;

        for(int i=0;i<arr1.size();i++)

        {

            for(int j=0;j<arr2.size();j++)

            {

                if(abs(arr1[i]-arr2[j])<=d)

                {

                    sum=sum+1;

                    break;

                }

            }

        }

        cout<<sum;

        return arr1.size()-sum;

    }

};
```