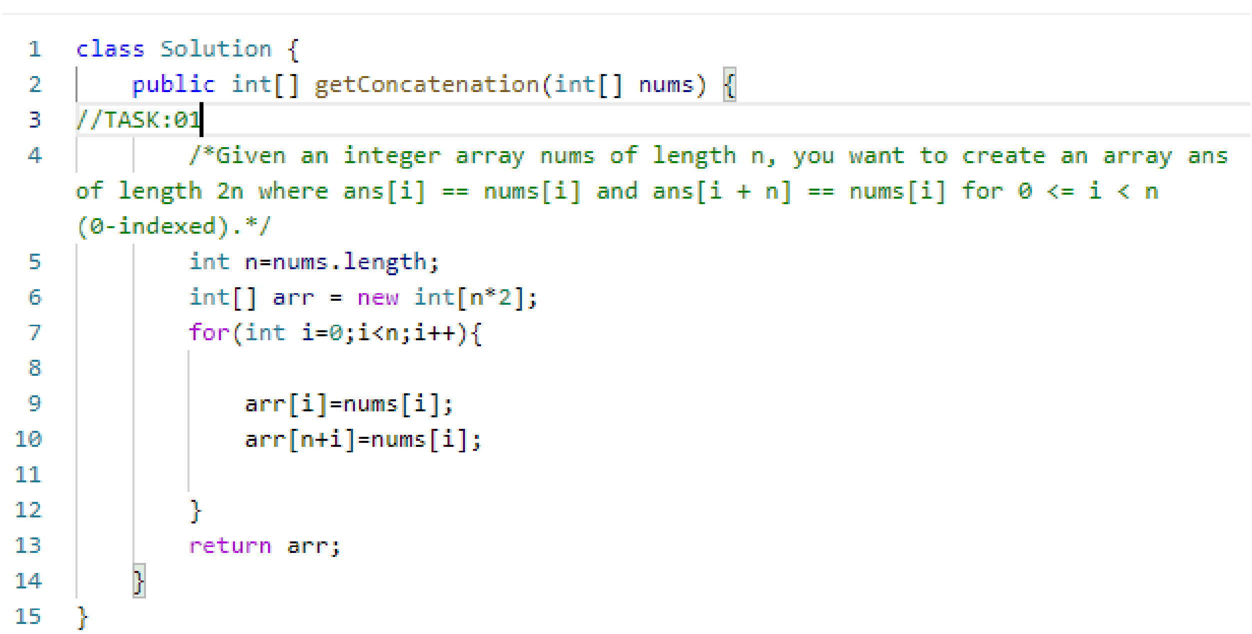
NAME:SANIA MEMON
ROLLNO: 0091
DATA STRUCTURE:LAB02
(ON LEETCODE)

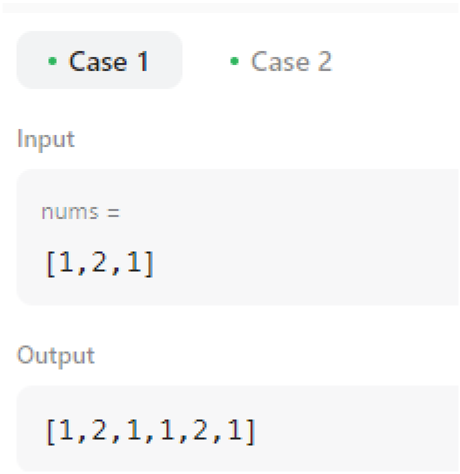
TASK01:



```
class Solution {
    public int[] getConcatenation(int[] nums) {
//TASK:01
        /*Given an integer array nums of length n, you want to create an array ans
of length 2n where ans[i] == nums[i] and ans[i + n] == nums[i] for 0 <= i < n
(0-indexed).*/
        int n=nums.length;
        int[] arr = new int[n*2];
        for(int i=0;i<n;i++){

            arr[i]=nums[i];
            arr[n+i]=nums[i];

        }
        return arr;
    }
}
```

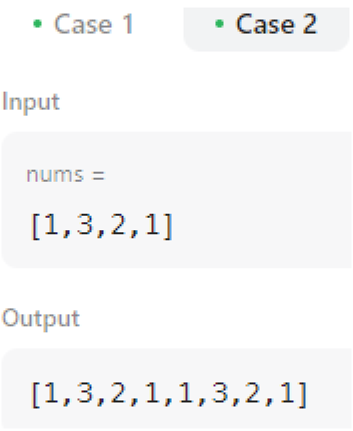


Input

```
nums =
[1,2,1]
```

Output

```
[1,2,1,1,2,1]
```



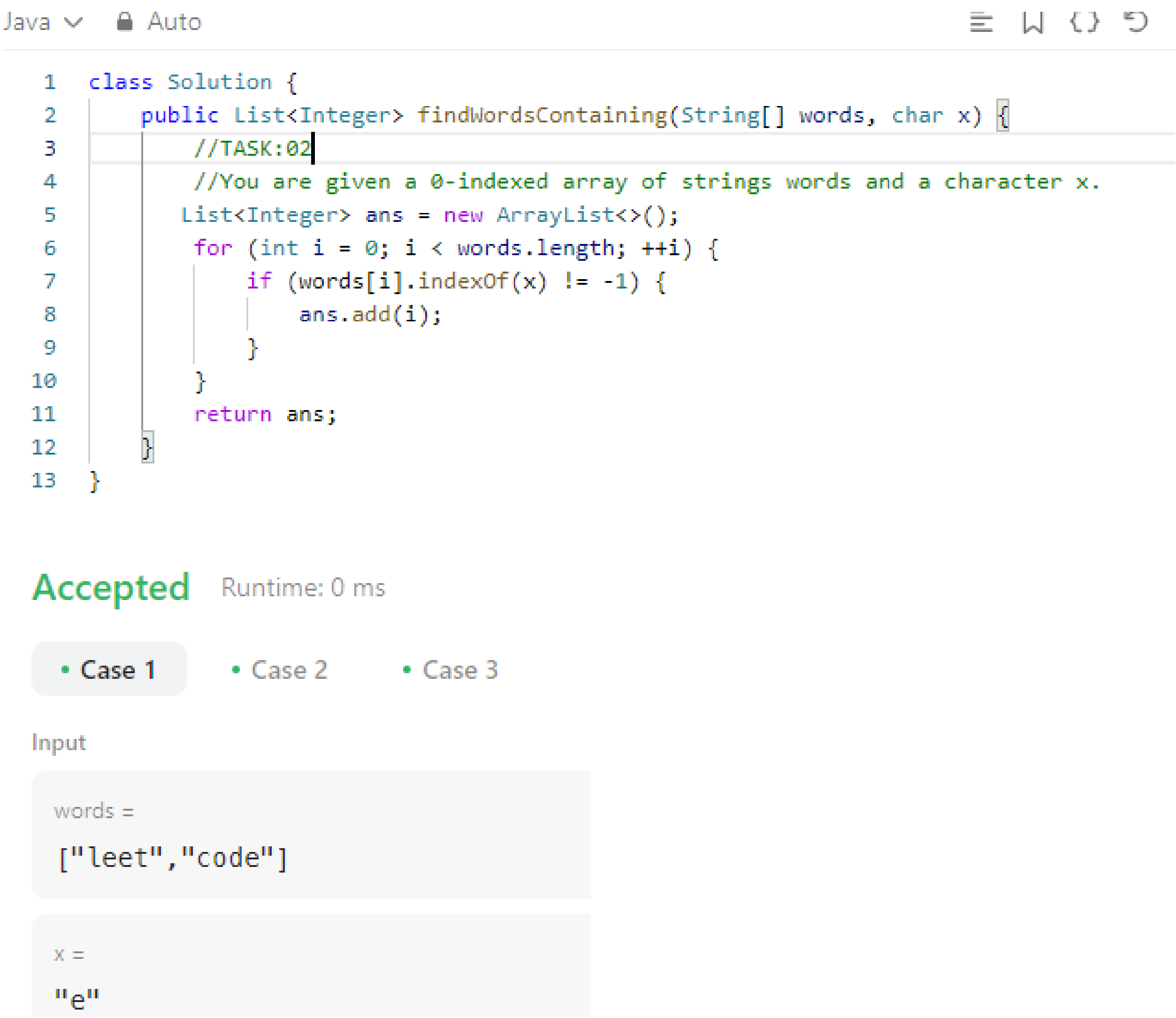
Input

```
nums =
[1,3,2,1]
```

Output

```
[1,3,2,1,1,3,2,1]
```

TASK 02:



```java
class Solution {
    public List<Integer> findWordsContaining(String[] words, char x) {
        //TASK:02
        //You are given a 0-indexed array of strings words and a character x.
       List<Integer> ans = new ArrayList<>();
        for (int i = 0; i < words.length; ++i) {
            if (words[i].indexOf(x) != -1) {
                ans.add(i);
            }
        }
        return ans;
    }
}
```

Accepted Runtime: 0 ms

Input

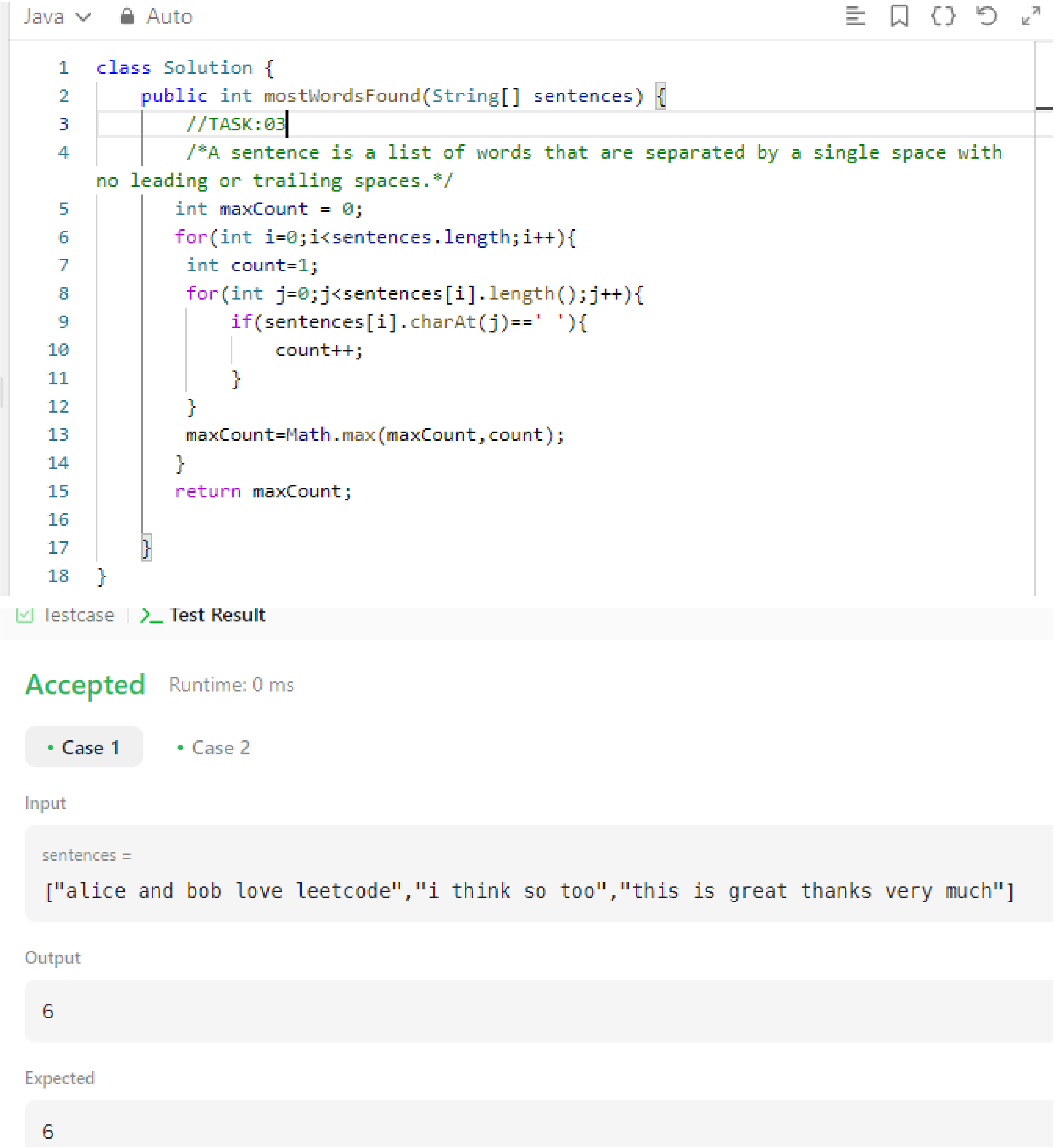
words =

```
["leet","code"]
```

x =

```
"e"
```

TASK 03:



```java
class Solution {
    public int mostWordsFound(String[] sentences) {
        //TASK:03
        /*A sentence is a list of words that are separated by a single space with no leading or trailing spaces.*/
       int maxCount = 0;
       for(int i=0;i<sentences.length;i++){
        int count=1;
        for(int j=0;j<sentences[i].length();j++){
            if(sentences[i].charAt(j)==' '){
                count++;
            }
        }
        maxCount=Math.max(maxCount,count);
       }
       return maxCount;

    }
}
```

Accepted Runtime: 0 ms

Input

sentences =

```
["alice and bob love leetcode","i think so too","this is great thanks very much"]
```

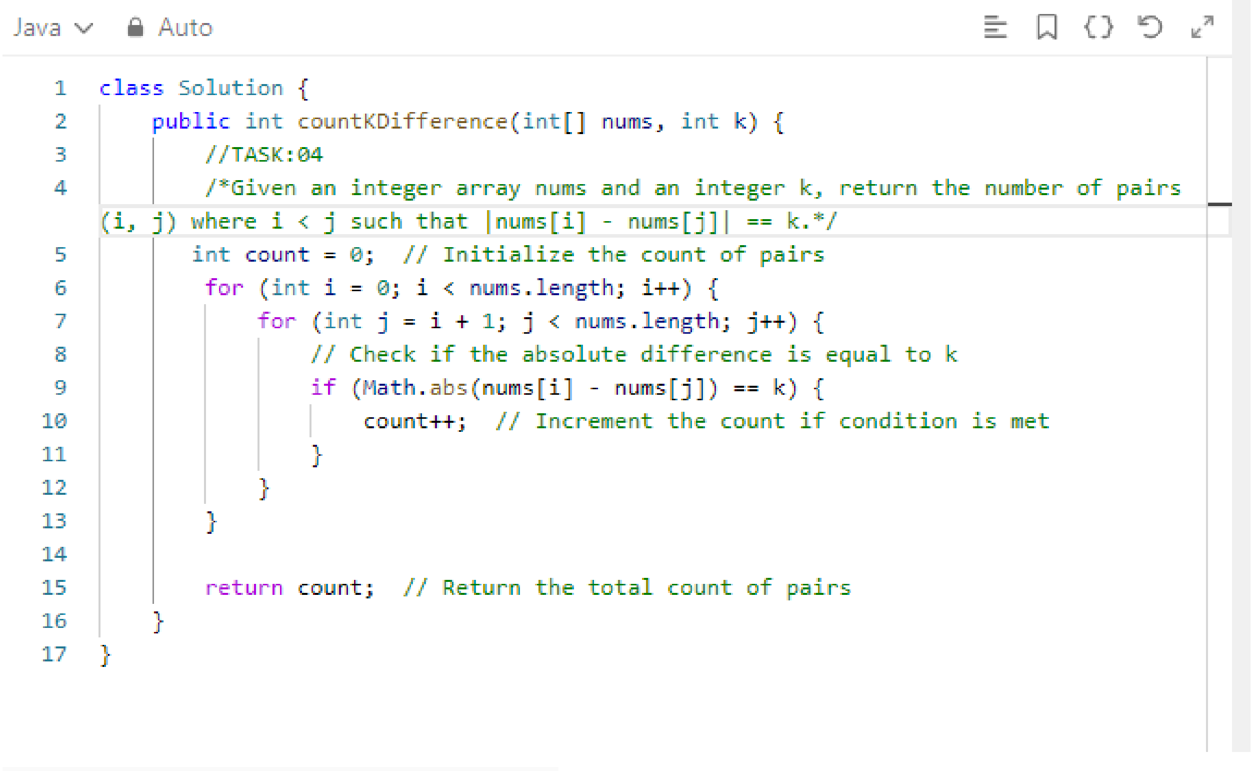
Output

```
6
```

Expected

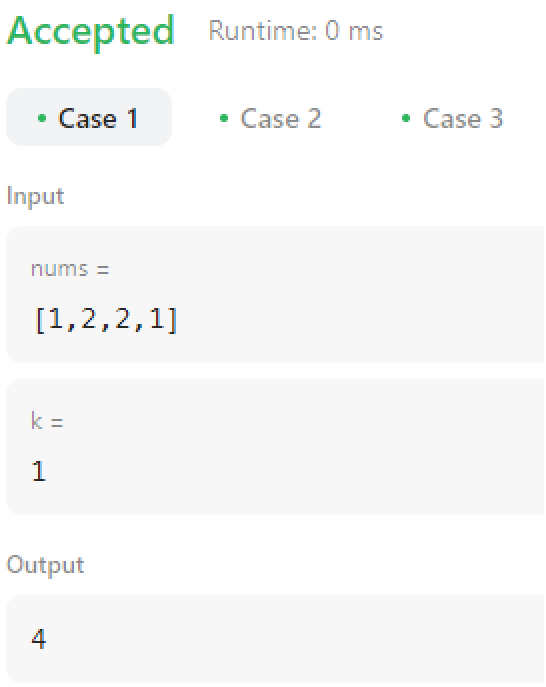
```
6
```

TASK 04:



```java
class Solution {
    public int countKDifference(int[] nums, int k) {
        //TASK:04
        /*Given an integer array nums and an integer k, return the number of pairs (i, j) where i < j such that |nums[i] - nums[j]| == k.*/
       int count = 0;  // Initialize the count of pairs
        for (int i = 0; i < nums.length; i++) {
            for (int j = i + 1; j < nums.length; j++) {
                // Check if the absolute difference is equal to k
                if (Math.abs(nums[i] - nums[j]) == k) {
                    count++;  // Increment the count if condition is met
                }
            }
        }

        return count;  // Return the total count of pairs
    }
}
```



Accepted Runtime: 0 ms

Input

nums =

[1,2,2,1]

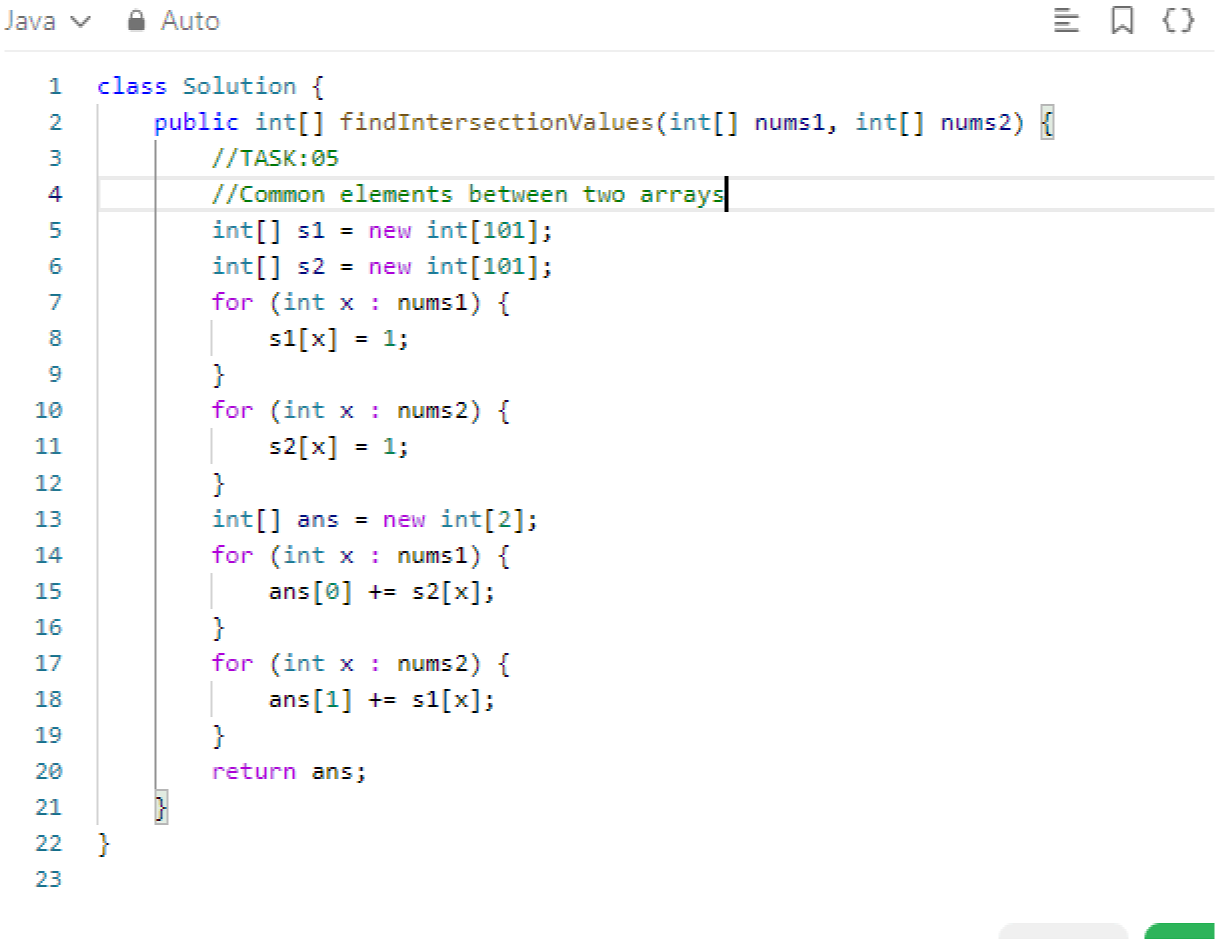
k =

1

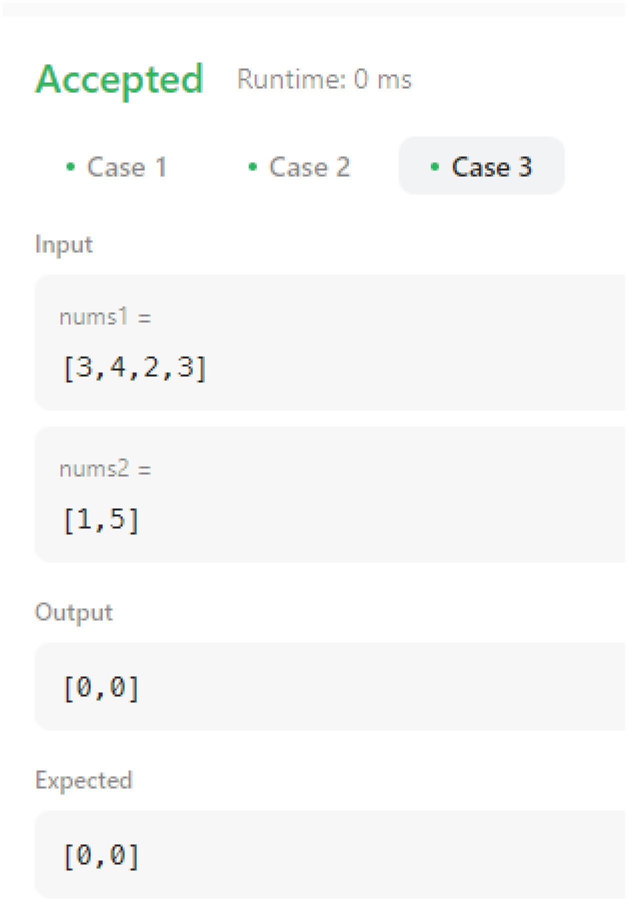
Output

4

TASK 05:



```java
class Solution {
    public int[] findIntersectionValues(int[] nums1, int[] nums2) {
        //TASK:05
        //Common elements between two arrays
        int[] s1 = new int[101];
        int[] s2 = new int[101];
        for (int x : nums1) {
            s1[x] = 1;
        }
        for (int x : nums2) {
            s2[x] = 1;
        }
        int[] ans = new int[2];
        for (int x : nums1) {
            ans[0] += s2[x];
        }
        for (int x : nums2) {
            ans[1] += s1[x];
        }
        return ans;
    }
}
```



Accepted Runtime: 0 ms

Input

nums1 =

```
[3,4,2,3]
```

nums2 =

```
[1,5]
```

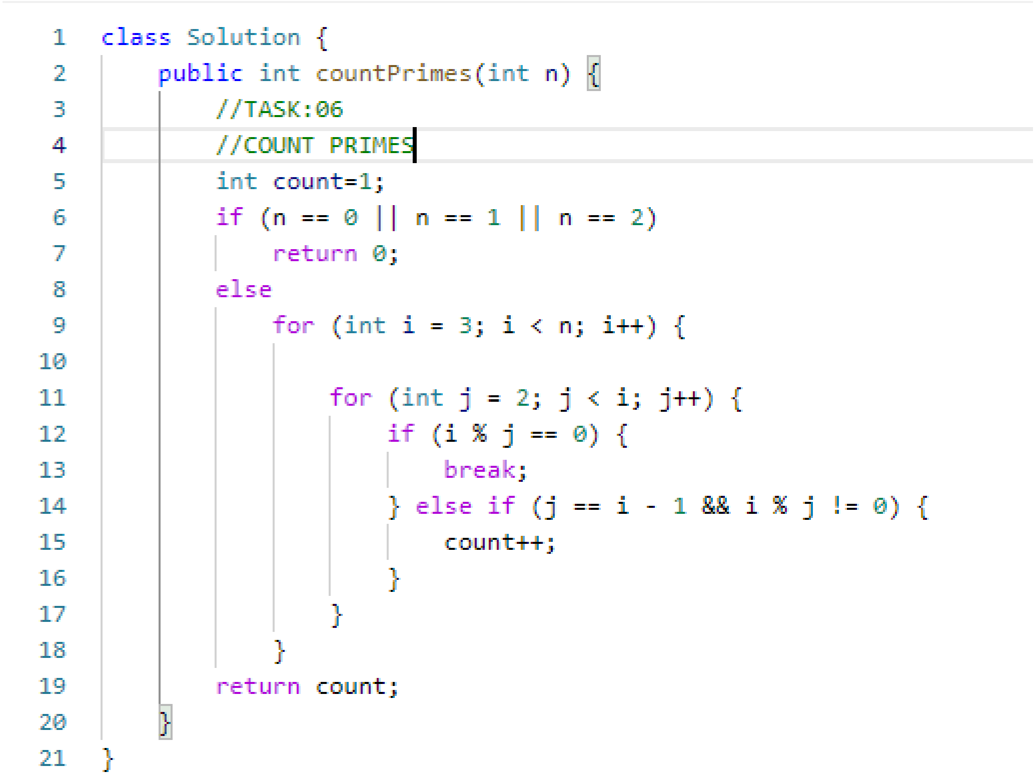
Output

```
[0,0]
```

Expected

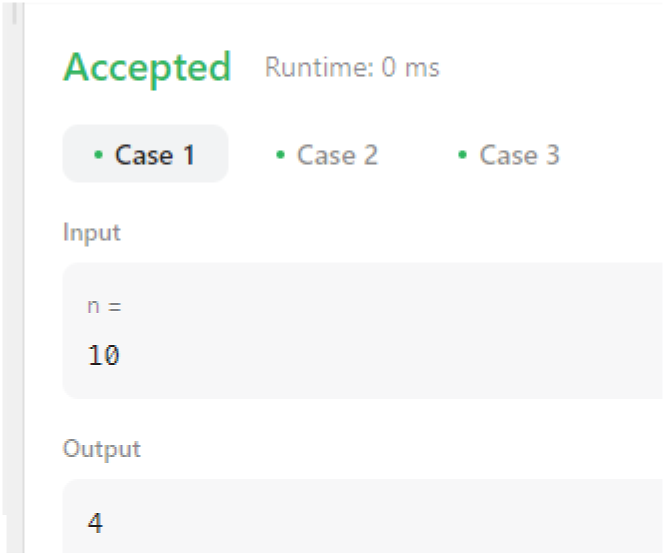
```
[0,0]
```

TASK 06:



```java
class Solution {
    public int countPrimes(int n) {
        //TASK:06
        //COUNT PRIMES
        int count=1;
        if (n == 0 || n == 1 || n == 2)
            return 0;
        else
            for (int i = 3; i < n; i++) {

                for (int j = 2; j < i; j++) {
                    if (i % j == 0) {
                        break;
                    } else if (j == i - 1 && i % j != 0) {
                        count++;
                    }
                }
            }
        return count;
    }
}
```



Accepted Runtime: 0 ms

Input

```
n =
10
```

Output

```
4
```