

## TASK 001:

Accepted

yasirashaikh submitted at Sep 12, 2024 22:58

Runtime: 1 ms | Beats 96.23%

Memory: 45.13 MB | Beats 70.60%

Code | Java

```
class Solution {
    public int[] getConcatenation(int[] nums) {
        int ans = nums.length;
        int[] arr = new int[ans*2];
        for(int i=0; i<ans; i++){
            arr[i] = nums[i];
        }
        return arr;
    }
}
```

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1

Input: nums = [1,2,1]

Output: [1,2,1,1,2,1]

Expected: [1,2,1,1,2,1]

## TASK 002:

Accepted

yasirashaikh submitted at Sep 12, 2024 23:06

Runtime: 1 ms | Beats 100.00%

Memory: 44.58 MB | Beats 92.53%

Code | Java

```
class Solution {
    public List<Integer> findWordsContaining(String[] words, char x) {
        List<Integer> list = new ArrayList<>();
        for(int i=0; i<words.length; i++){
            String word = words[i];
            for(int j=0; j<word.length(); j++){
                if(word.charAt(j)==x){
                    list.add(i);
                    break;
                }
            }
        }
        return list;
    }
}
```

Testcase | Test Result

Case 1 Case 2 Case 3

Input: words = [\"abc\", \"bcd\", \"aaaa\", \"cbc\"]

x = \"z\"

Output: []

## TASK 003:

Accepted

yasirashalkh submitted at Sep 12, 2024 23:10

Runtime: 3 ms | Beats: 86.89% | Memory: 44.07 MB | Beats: 41.55%

Bar chart showing runtime distribution for the 'mostWordsFound' problem. The x-axis represents runtime in milliseconds (1ms to 10ms), and the y-axis represents the percentage of submissions (0% to 75%). The distribution is heavily skewed towards 3ms, with approximately 75% of submissions achieving this runtime.

Code (Java)

```
class Solution {
    public int mostWordsFound(String[] sentences) {
        int ans = 0;
        for(int i = 0; i < sentences.length; i++){
            String s = sentences[i];
            String[] arr = s.strip().split(" ");
            ans = arr.length;
        }
        return ans;
    }
}
```

Testcase

Accepted Runtime: 0 ms

Case 1

Input: sentences = ["alice and bob love leetcode", "i think so too", "this is great thanks very much"]

Output: 6

Task 005:

Accepted

yasirashalkh submitted at Sep 12, 2024 23:24

Runtime: 5 ms | Beats: 79.40% | Memory: 45.43 MB | Beats: 51.51%

Bar chart showing runtime distribution for the 'findIntersectionValues' problem. The x-axis represents runtime in milliseconds (1ms to 21ms), and the y-axis represents the percentage of submissions (0% to 40%). The distribution is skewed towards 5ms, with approximately 35% of submissions achieving this runtime.

Code (Java)

```
class Solution {
    public int[] findIntersectionValues(int[] nums1, int[] nums2) {
        int ans1 = 0;
        int ans2 = 0;
        for (int i = 0; i < nums1.length; i++) {
            for (int j = 0; j < nums2.length; j++) {
                if (nums1[i] == nums2[j]) {
                    ans1++;
                    break;
                }
            }
        }
        for (int i = 0; i < nums2.length; i++) {
            for (int j = 0; j < nums1.length; j++) {
                if (nums2[i] == nums1[j]) {
                    ans2++;
                }
            }
        }
        return new int[] {ans1, ans2};
    }
}
```

Testcase

Accepted Runtime: 0 ms

Case 1

Input: nums1 = [2,3,2], nums2 = [1,2]

Output: [2,1]

Task 006:

DescriptionEditorialSolutionsAcceptedSubmissions

All Submissions

Accepted

yasirashikh submitted at Sep 12, 2024 23:29

EditorialSolution


Runtime

118 ms | Beats 23.84%

Analyze Complexity

Memory

49.64 MB | Beats 23.63%



Code | Java

```
class Solution {
    public int countPrimes(int n) {
        if(n <= 2) return 0;
        int count = 0;
        boolean[] isPrime = new boolean[n];
        for(int i = 2; i < n; i++){
            isPrime[i] = true;
        }
        for(int i = 2; i < n; i++){
            if(isPrime[i]){
                count++;
            }
        }
        return count;
    }
}
```

Code

JavaAuto

```
1 class Solution {
2     public int countPrimes(int n) {
3         if(n <= 2) return 0;
4         int count = 0;
5         boolean[] isPrime = new boolean[n];
6         for(int i = 2; i < n; i++){
7             isPrime[i] = true;
8         }
9         for(int i = 2; i < n; i++){
10             if(isPrime[i]){
11                 count++;
12             }
13         }
14         return count;
15     }
16 }
17
```

SavedLn 17, Col 8

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3

Input

n = 10

Output

4

Expected