Description | Accepted | Editorial | Solutions | Submissi

1295. Find Numbers with Even Number of Digits

Solved

Easy | Topics | Companies | Hint

Given an array `nums` of integers, return how many of them contain an **even number** of digits.

Example 1:

```
Input: nums = [12,345,2,6,7896]
Output: 2
Explanation:
12 contains 2 digits (even number of digits).
345 contains 3 digits (odd number of digits).
2 contains 1 digit (odd number of digits).
6 contains 1 digit (odd number of digits).
7896 contains 4 digits (even number of digits).
Therefore only 12 and 7896 contain an even number of digits.
```

Example 2:

```
Input: nums = [555,901,482,1771]
Output: 1
Explanation:
Only 1771 contains an even number of digits.
```

Constraints:

2.9K | 204 | 43 Online

Code

Java | Auto

```java
class Solution {
    public int findNumbers(int[] nums) {
        int count = 0;

        for (int n : nums) {
            int digits = String.valueOf(n).length();
            if (digits % 2 == 0) {
                count++;
            }
        }

        return count;
    }
}
```

Saved | Ln 15, Col 1

Testcase | Test Result

You must run your code first

Description | Accepted | Editorial | Solutions | Submissi

1491. Average Salary Excluding the Minimum and Maximum Salary

Solved

Easy | Topics | Companies | Hint

You are given an array of **unique** integers `salary` where `salary[i]` is the salary of the i^{th} employee.

Return *the average salary of employees excluding the minimum and maximum salary*. Answers within 10^{-5} of the actual answer will be accepted.

Example 1:

```
Input: salary = [4000,3000,1000,2000]
Output: 2500.00000
Explanation: Minimum salary and maximum salary are 1000 and 4000 respectively.
Average salary excluding minimum and maximum salary is (2000+3000) / 2 = 2500
```

Example 2:

```
Input: salary = [1000,2000,3000]
Output: 2000.00000
Explanation: Minimum salary and maximum salary are 1000 and 3000 respectively.
Average salary excluding minimum and maximum salary is (2000) / 1 = 2000
```

2.3K | 86 | 16 Online

Code

Java | Auto

```java
class Solution {
    public double average(int[] salary) {
        int min = Integer.MAX_VALUE;
        int max = Integer.MIN_VALUE;
        int sum = 0;

        for (int s : salary) {
            sum += s;
            if (s < min) min = s;
            if (s > max) max = s;
        }

        return (double)(sum - min - max) / (salary.length - 2);
    }
}
```

Saved | Ln 16, Col 1

Testcase | Test Result

salary =

[4000,3000,1000,2000]

Output

2500.00000

Expected

2500.00000

Contribute a testcase

Description | Accepted | Editorial | Solutions

1480. Running Sum of 1d Array

Solved

Easy | Topics | Companies | Hint

Given an array `nums`. We define a running sum of an array as `runningSum[i] = sum(nums[0]…nums[i])`.

Return the running sum of `nums`.

Example 1:

```
Input: nums = [1,2,3,4]
Output: [1,3,6,10]
Explanation: Running sum is obtained as follows: [1, 1+2, 1+2+3, 1+2+3+4].
```

Example 2:

```
Input: nums = [1,1,1,1,1]
Output: [1,2,3,4,5]
Explanation: Running sum is obtained as follows: [1, 1+1, 1+1+1, 1+1+1+1, 1+1+1+1+1].
```

Example 3:

```
Input: nums = [3,1,2,10,1]
Output: [3,4,6,16,17]
```

Constraints:

Code

Java | Auto

```java
class Solution {
    public int[] runningSum(int[] nums) {
        for(int i=1; i< nums.length;i++){
            nums[i] = nums[i] + nums[i-1];
        }
        return nums;
    }
}
```

Saved

Ln 3, Col 24

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

nums =
[1,2,3,4]

Output

[1,3,6,10]