Mock Test > contato@guifr.com.br

Full Name: GUILHERME FERNANDES Email: contato@guifr.com.br Test Name: **Mock Test** 22 May 2022 01:58:12 IST Taken On: Time Taken: 24 min 19 sec/ 30 min Invited by: Ankush 22 May 2022 01:58:05 IST Invited on: Skills Score: Tags Score: Algorithms 105/105 Core CS 105/105 Data Structures 105/105 Easy 105/105 LCM 105/105 Least Common Multiple 105/105 105/105 Math gcd 105/105 greatest common divisor 105/105 problem-solving 105/105 sets 105/105

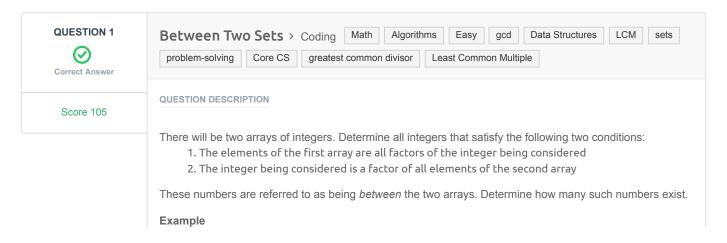
100% 105/105

scored in **Mock Test** in 24 min 19 sec on 22 May 2022 01:58:12 IST

Recruiter/Team Comments:

No Comments.





```
a = [2, 6] \ b = [24, 36]
```

There are two numbers between the arrays: 6 and 12.

6%2 = 0, 6%6 = 0, 24%6 = 0 and 36%6 = 0 for the first value.

12%2 = 0, 12%6 = 0 and 24%12 = 0, 36%12 = 0 for the second value. Return 2.

Function Description

Complete the *getTotalX* function in the editor below. It should return the number of integers that are betwen the sets.

getTotalX has the following parameter(s):

- int a[n]: an array of integers
- int b[m]: an array of integers

Returns

• int: the number of integers that are between the sets

Input Format

The first line contains two space-separated integers, n and m, the number of elements in arrays a and b. The second line contains n distinct space-separated integers a[i] where $0 \le i < n$.

The third line contains m distinct space-separated integers b[j] where $0 \leq j < m$.

Constraints

- $1 \le n, m \le 10$
- $1 \le a[i] \le 100$
- $1 \le b[j] \le 100$

Sample Input

```
2 3
2 4
16 32 96
```

Sample Output

3

Explanation

2 and 4 divide evenly into 4, 8, 12 and 16.

- 4, 8 and 16 divide evenly into 16, 32, 96.
- 4, 8 and 16 are the only three numbers for which each element of a is a factor and each is a factor of all elements of b.

CANDIDATE ANSWER

Language used: Java 8

```
1 class Result {
2
3    /*
4     * Complete the 'getTotalX' function below.
5     *
6     * The function is expected to return an INTEGER.
7     * The function accepts following parameters:
8     * 1. INTEGER_ARRAY a
9     * 2. INTEGER_ARRAY b
10     */
```

```
public static int getTotalX(List<Integer> a, List<Integer> b) {
           System.out.println(a);
           System.out.println(b);
           int result = 0;
           Collections.sort(b);
           int bigNumber = b.get(b.size() - 1);
           List<Integer> factors = new ArrayList<>();
           for(int n = 1; n <= Math.sqrt(bigNumber);n++) {</pre>
               if(bigNumber % n == 0) {
                   int quo = bigNumber / n;
                   factors.add(n);
                   if(n != quo)
                       factors.add(quo);
           }
           for(int between : factors){
               boolean satisfy = true;
               for(int number : a) {
                    if(between % number != 0 || !satisfy) {
34
                       satisfy = false;
                       break;
                   }
               }
               for(int number : b) {
                    if(number % between != 0 || !satisfy) {
40
                       satisfy = false;
                       break;
                   }
44
               }
46
               if(satisfy){
                   System.out.println(between);
                    result++;
               }
           }
           System.out.println(factors);
           return result;
58 }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.148 sec	29.7 KB
Testcase 2	Easy	Hidden case	Success	15	0.1487 sec	30.2 KB
Testcase 3	Easy	Hidden case	Success	15	0.1969 sec	29.9 KB
Testcase 4	Easy	Hidden case	Success	15	0.1444 sec	30.1 KB
Testcase 5	Easy	Hidden case	Success	15	0.127 sec	29.6 KB
Testcase 6	Easy	Hidden case	Success	15	0.2581 sec	30.1 KB
Testcase 7	Easy	Hidden case	Success	15	0.1792 sec	29.9 KB

Testcase 8	Easy	Hidden case	Success	15	0.1218 sec	30 KB	
Testcase 9	Easy	Sample case	Success	0	0.1551 sec	29.8 KB	
No Comments							

PDF generated at: 21 May 2022 20:54:18 UTC