Mock Test > contato@guifr.com.br

Full Name:	GUILHERME FERNANDES
Email:	contato@guifr.com.br
Test Name:	**Mock Test**
Taken On:	22 May 2022 01:58:12 IST
Time Taken:	24 min 19 sec/ 30 min
Invited by:	Ankush
Invited on:	22 May 2022 01:58:05 IST
Skills Score:	

100%

105/105

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

Tags Score:

Algorithms	105/105
Core CS	105/105
Data Structures	105/105
Easy	105/105
LCM	105/105
Least Common Multiple	105/105
Math	105/105
gcd	105/105
greatest common divisor	105/105
problem-solving	105/105
sets	105/105

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	**Between Two Sets** > **Coding**	24 min 1 sec	105/ 105	✓

QUESTION 1

✓

Correct Answer

Score 105

Between Two Sets > Coding

Math | Algorithms | Easy | gcd | Data Structures | LCM | sets | problem-solving | Core CS | greatest common divisor | Least Common Multiple

QUESTION DESCRIPTION

There will be two arrays of integers. Determine all integers that satisfy the following two conditions:

1. The elements of the first array are all factors of the integer being considered
2. The integer being considered is a factor of all elements of the second array

These numbers are referred to as being *between* the two arrays. Determine how many such numbers exist.

Example

$a = [2, 6]$
$b = [24, 36]$

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

Constraints

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

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**

```
class Result {

    /*
     * Complete the 'getTotalX' function below.
     *
     * The function is expected to return an INTEGER.
     * The function accepts following parameters:
     *  1. INTEGER_ARRAY a
     *  2. INTEGER_ARRAY b
     */
```

```java

    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++){
            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){
                    satisfy = false;
                    break;
                }
            }

            for(int number : b){
                if(number % between != 0 || !satisfy){
                    satisfy = false;
                    break;
                }
            }

            if(satisfy){
                System.out.println(between);
                result++;
            }
        }

        System.out.println(factors);

        return result;

    }

}


```

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