Full Name:	Remi Chartier
Email:	remipr.chartier@gmail.com
Test Name:	**Mock Test**
Taken On:	15 Nov 2021 23:21:22 IST
Time Taken:	29 min 46 sec/ 30 min
Contact Number:	+14084751573
Linkedin:	http://www.linkedin.com/in/remichartier
Invited by:	Ankush
Invited on:	15 Nov 2021 23:21:12 IST
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 Math 105/105 gcd 105/105 greatest common divisor 105/105 problem-solving 105/105 sets 105/105

100%

105/105

scored in **Mock Test** in 29 min 46 sec on 15 Nov 2021 23:21:22 IST

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review.

	Question Description	Time Taken	Score	Status
Q1	Between Two Sets > **Coding**	28 min 8 sec	105/ 105	!

QUESTION 1

Needs Review

Between Two Sets > Coding

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

Score 105

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: **Python 3**

```
#
# Complete the 'getTotalX' function below.
```

```
# 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
#

def getTotalX(a, b):
    # Write your code here
    # 1. elements 1st array are all factors of integer being considered
    # 2. integer being considered is a factor of all elements of the 2nd
array.
        # 1. elements 1st array are all factors of integer being considered
    # 2. integer being considered is a factor of all elements of the 2nd
array.
    max_elem = max(b)
    min_elem = max(a)
    count = 0
    for integer in range(min_elem,max_elem + 1):
        consider_elem = True
        for i in a:
            if integer%i != 0:
                consider_elem = False
                break;
        if not consider_elem:
            continue
        for i in b:
            if i%integer != 0:
                consider_elem = False
                break;
        # here, integer passed all the tests --> count it.
        if consider_elem:
            count += 1
    return count
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.0444 sec	9.48 KB
Testcase 2	Easy	Hidden case	Success	15	0.0435 sec	9.48 KB
Testcase 3	Easy	Hidden case	Success	15	0.0449 sec	9.38 KB
Testcase 4	Easy	Hidden case	Success	15	0.0496 sec	9.39 KB
Testcase 5	Easy	Hidden case	Success	15	0.0441 sec	9.45 KB
Testcase 6	Easy	Hidden case	Success	15	0.0632 sec	9.41 KB
Testcase 7	Easy	Hidden case	Success	15	0.0409 sec	9.27 KB
Testcase 8	Easy	Hidden case	Success	15	0.0483 sec	9.53 KB
Testcase 9	Easy	Sample case	Success	0	0.0402 sec	9.48 KB

No Comments