Full Name:	Musab Oguz
Email:	doliatius@protonmail.com
Test Name:	**Mock Test**
Taken On:	18 Jan 2024 17:30:43 IST
Time Taken:	24 min 17 sec/ 30 min
Linkedin:	https://www.linkedin.com/in/musab-oguz-68990a200/
Invited by:	Ankush
Invited on:	18 Jan 2024 17:30:37 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 24 min 17 sec on 18 Jan 2024 17:30:43 IST

Recruiter/Team Comments:

No Comments.

Question Description	Time Taken	Score	Status
Q1 **Between Two Sets** > **Coding**	24 min 7 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: **C++14**

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

```cpp
int getTotalX(vector<int> a, vector<int> b) {

    sort(a.begin(), a.end());
    sort(b.begin(), b.end());

    int limit = b.at(0);
    int first = a.at(0);
    int start = 0;

    vector<int> factors;
    vector<int> ret_vec;

    while (start < limit){
        bool is_good = true;
        start += first;
        for (int i = 0; i<a.size(); i++){
            if (start % a.at(i) > 0){
                is_good = false;
            }
        }
        if (is_good) {factors.push_back(start);}
    }

    for (int i = 0; i<factors.size(); i++){
        bool is_good = true;
        int divide = factors.at(i);

        for (int ii = 0; ii<b.size(); ii++){
            if (b.at(ii) % divide > 0){
                is_good = false;
            }
        }

        if (is_good) {ret_vec.push_back(divide);}
    }

    return ret_vec.size();
}

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.0064 sec	8.82 KB
Testcase 2	Easy	Hidden case	Success	15	0.008 sec	8.77 KB
Testcase 3	Easy	Hidden case	Success	15	0.0064 sec	8.87 KB
Testcase 4	Easy	Hidden case	Success	15	0.0066 sec	8.82 KB
Testcase 5	Easy	Hidden case	Success	15	0.0075 sec	8.93 KB
Testcase 6	Easy	Hidden case	Success	15	0.0075 sec	8.88 KB
Testcase 7	Easy	Hidden case	Success	15	0.0064 sec	8.82 KB
Testcase 8	Easy	Hidden case	Success	15	0.0052 sec	8.78 KB
Testcase 9	Easy	Sample case	Success	0	0.0052 sec	8.87 KB

No Comments