Mock Test › mbkara0106@gmail.com

Full Name:	Bugrahan Kara
Email:	mbkara0106@gmail.com
Test Name:	**Mock Test**
Taken On:	3 Apr 2022 13:00:02 IST
Time Taken:	27 min 25 sec/ 30 min
Resume:	https://hackerrank-resumes.s3.amazonaws.com/14313602/2d5wcNSlVi9yCF6eHTOIaQiPWrOmbN2f_3-qP995O3kQsQIIk7TcmvhCrNQrZkSyBA/Muhammet_Bugrahan_KARA_CV6.pdf
Invited by:	Ankush
Invited on:	3 Apr 2022 12:50:51 IST
Skills Score:	

100%

105/105

scored in **Mock Test** in 27 min 25 sec on 3 Apr 2022 13:00:02 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**	27 min 18 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**

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

```
 * The function accepts following parameters:
 *  1. INTEGER_ARRAY a
 *  2. INTEGER_ARRAY b
 */

int getTotalX(vector<int> a, vector<int> b) {
    int result;
    int maxA= *max_element(a.begin(),a.end());
    int minB= *min_element(b.begin(),b.end());
    vector<int> samples;
    vector<int> samples2;
    for(int i=maxA;i<=minB;i++){
       int divided=0;
       for(auto x:a){
           if(i%x==0)
             divided++;
        }
        if(divided==a.size())
            samples.push_back(i);
    }
    //cout<<samples.size()<<endl;


    for(int i=0;i<samples.size();i++){
        int divided=0;
        for(auto y:b){
            if(y%samples[i]==0)
                divided++;
        }
        if(divided==b.size()){
            samples2.push_back(samples[i]);
        }
    }
    result=samples2.size();

    return result;
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✓ Success	0	0.0253 sec	9.11 KB
Testcase 2	Easy	Hidden case	✓ Success	15	0.0293 sec	8.92 KB
Testcase 3	Easy	Hidden case	✓ Success	15	0.019 sec	8.95 KB
Testcase 4	Easy	Hidden case	✓ Success	15	0.0208 sec	8.76 KB
Testcase 5	Easy	Hidden case	✓ Success	15	0.0287 sec	8.93 KB
Testcase 6	Easy	Hidden case	✓ Success	15	0.0218 sec	8.86 KB
Testcase 7	Easy	Hidden case	✓ Success	15	0.0242 sec	8.99 KB
Testcase 8	Easy	Hidden case	✓ Success	15	0.0226 sec	9.03 KB
Testcase 9	Easy	Sample case	✓ Success	0	0.0227 sec	8.96 KB

No Comments