



Full Name:	MD MAFUJUL HASAN
Email:	mdtonmoy13.mt@gmail.com
Test Name:	Mock Test
Taken On:	14 May 2023 14:41:15 IST
Time Taken:	19 min 28 sec/ 90 min
Invited by:	Ankush
Invited on:	14 May 2023 14:41:04 IST
Skills Score:	
Tags Score:	<div>Algorithms280/280</div> <div>Core CS280/280</div> <div>Data Structures105/105</div> <div>Easy280/280</div> <div>LCM105/105</div> <div>Least Common Multiple105/105</div> <div>Math105/105</div> <div>Problem Solving105/105</div> <div>Strings175/175</div> <div>gcd105/105</div> <div>greatest common divisor105/105</div> <div>problem-solving280/280</div> <div>sets105/105</div>

100%  
280/280

scored in **Mock Test** in 19 min  
28 sec on 14 May 2023 14:41:15  
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	Palindrome Index > Coding	2 min 44 sec	105/ 105	⚠
Q2	Between Two Sets > Coding	3 min 31 sec	105/ 105	✅
Q3	Anagram > Coding	4 min 8 sec	70/ 70	✅

## QUESTION 1



Needs Review

Score 105

## Palindrome Index &gt; Coding

Strings

Algorithms

Easy

problem-solving

Core CS

Problem Solving

## QUESTION DESCRIPTION

Given a string of lowercase letters in the range `ascii[a-z]`, determine the index of a character that can be removed to make the string a **palindrome**. There may be more than one solution, but any will do. If the word is already a palindrome or there is no solution, return `-1`. Otherwise, return the index of a character to remove.

## Example

 $s = \text{"bcbc"}$ 

Either remove 'b' at index **0** or 'c' at index **3**.

## Function Description

Complete the `palindromeIndex` function in the editor below.

`palindromeIndex` has the following parameter(s):

- *string s*: a string to analyze

## Returns

- *int*: the index of the character to remove or `-1`

## Input Format

The first line contains an integer  $q$ , the number of queries.

Each of the next  $q$  lines contains a query string  $s$ .

## Constraints

- $1 \leq q \leq 20$
- $1 \leq \text{length of } s \leq 10^5 + 5$
- All characters are in the range `ascii[a-z]`.

## Sample Input

STDIN	Function
3	$q = 3$
aaab	$s = \text{'aaab'}$ (first query)
baa	$s = \text{'baa'}$ (second query)
aaa	$s = \text{'aaa'}$ (third query)

## Sample Output

```
3
0
-1
```

## Explanation

Query 1: "aaab"

Removing 'b' at index **3** results in a palindrome, so return **3**.

Query 2: "baa"

Removing 'b' at index **0** results in a palindrome, so return **0**.

Query 3: "aaa"
















This string is already a palindrome, so return `-1`. Removing any one of the characters would result in a palindrome, but this test comes first.

Note: The custom checker logic for this challenge is available [here](#).

## CANDIDATE ANSWER

Language used: **Python 3**

```
1 #
2 # Complete the 'palindromeIndex' function below.
3 #
4 # The function is expected to return an INTEGER.
5 # The function accepts STRING s as parameter.
6 #
7
8 def find_mismatching_pair(s):
9     i = 0
10    j = len(s) - 1
11    while i < j and s[i] == s[j]:
12        i += 1
13        j -= 1
14    return i, j
15
16
17 def is_palindrome(s):
18     i, j = find_mismatching_pair(s)
19     return j <= i
20
21
22 def palindromeIndex(s):
23     # Write your code here
24     i, j = find_mismatching_pair(s)
25     return -1 if j <= i else i if is_palindrome(s[i + 1: j + 1]) else j
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	 Success	0	0.0512 sec	9.16 KB
Testcase 2	Medium	Hidden case	 Success	5	0.106 sec	9.2 KB
Testcase 3	Medium	Hidden case	 Success	5	0.0838 sec	9.2 KB
Testcase 4	Medium	Hidden case	 Success	5	0.0384 sec	9.34 KB
Testcase 5	Medium	Hidden case	 Success	5	0.059 sec	9.27 KB
Testcase 6	Medium	Hidden case	 Success	5	0.0854 sec	9.59 KB
Testcase 7	Medium	Hidden case	 Success	5	0.126 sec	9.62 KB
Testcase 8	Medium	Hidden case	 Success	5	0.1264 sec	9.73 KB
Testcase 9	Hard	Hidden case	 Success	10	0.0516 sec	9.64 KB
Testcase 10	Hard	Hidden case	 Success	10	0.0704 sec	9.53 KB
Testcase 11	Hard	Hidden case	 Success	10	0.0597 sec	9.54 KB
Testcase 12	Hard	Hidden case	 Success	10	0.0884 sec	9.18 KB
Testcase 13	Hard	Hidden case	 Success	10	0.0922 sec	9.51 KB
Testcase 14	Hard	Hidden case	 Success	10	0.0829 sec	9.5 KB
Testcase 15	Hard	Hidden case	 Success	10	0.0626 sec	9.64 KB

No Comments

## QUESTION 2



Correct Answer

Score 105

## Between Two Sets

&gt; 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 between 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**

```
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 #
10
11 def getTotalX(a, b):
12     # Write your code here
13     answer = 0
14
15     for i in range(1, 101):
16         if all(i % x == 0 for x in a) and all(x % i == 0 for x in b):
17             answer += 1
18     return answer
19
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✔ Success	0	0.0532 sec	9.21 KB
Testcase 2	Easy	Hidden case	✔ Success	15	0.1163 sec	9.09 KB
Testcase 3	Easy	Hidden case	✔ Success	15	0.1041 sec	9.29 KB
Testcase 4	Easy	Hidden case	✔ Success	15	0.0932 sec	9.24 KB
Testcase 5	Easy	Hidden case	✔ Success	15	0.104 sec	9.26 KB
Testcase 6	Easy	Hidden case	✔ Success	15	0.0735 sec	9.42 KB
Testcase 7	Easy	Hidden case	✔ Success	15	0.0593 sec	9.18 KB
Testcase 8	Easy	Hidden case	✔ Success	15	0.1194 sec	9.37 KB
Testcase 9	Easy	Sample case	✔ Success	0	0.0974 sec	9.35 KB

No Comments

## QUESTION 3



Correct Answer

Score 70

## Anagram &gt; Coding

Strings

Algorithms

Easy

problem-solving

Core CS

## QUESTION DESCRIPTION

Two words are *anagrams* of one another if their letters can be rearranged to form the other word.

Given a string, split it into two contiguous substrings of equal length. Determine the minimum number of characters to change to make the two substrings into anagrams of one another.

**Example*****s* = abccde**

Break *s* into two parts: 'abc' and 'cde'. Note that all letters have been used, the substrings are contiguous and their lengths are equal. Now you can change 'a' and 'b' in the first substring to 'd' and 'e' to have 'dec' and 'cde' which are anagrams. Two changes were necessary.

### Function Description

Complete the *anagram* function in the editor below.

*anagram* has the following parameter(s):

- *string s*: a string

### Returns

- *int*: the minimum number of characters to change or -1.

### Input Format

The first line will contain an integer, *q*, the number of test cases.

Each test case will contain a string *s*.

### Constraints

- $1 \leq q \leq 100$
- $1 \leq |s| \leq 10^4$
- *s* consists only of characters in the range `ascii[a-z]`.

### Sample Input

```
6
aaabbb
ab
abc
mnop
xyyx
xaxbbxx
```

### Sample Output

```
3
1
-1
2
0
1
```

### Explanation

*Test Case #01*: We split *s* into two strings *S1*='aaa' and *S2*='bbb'. We have to replace all three characters from the first string with 'b' to make the strings anagrams.

*Test Case #02*: You have to replace 'a' with 'b', which will generate "bb".

*Test Case #03*: It is not possible for two strings of unequal length to be anagrams of one another.

*Test Case #04*: We have to replace both the characters of first string ("mn") to make it an anagram of the other one.

*Test Case #05*: *S1* and *S2* are already anagrams of one another.

*Test Case #06*: Here *S1* = "xaxb" and *S2* = "bbxx". You must replace 'a' from *S1* with 'b' so that *S1* = "xbxb".

### CANDIDATE ANSWER















Language used: **Python 3**



```
1 #!/bin/python3
2
3 import math
4 import os
```

```

5 import random
6 import re
7 import sys
8 from collections import Counter
9
10
11
12 #
13 # Complete the 'anagram' function below.
14 #
15 # The function is expected to return an INTEGER.
16 # The function accepts STRING s as parameter.
17 #
18
19 def anagram(s):
20     # Write your code here
21     length_s = len(s)
22     if length_s % 2:
23         return -1
24     mid = length_s // 2
25     a = Counter(s[:mid])
26     b = Counter(s[mid:])
27
28     return mid - sum((a & b).values())
29
30 if __name__ == '__main__':
31     fptr = open(os.environ['OUTPUT_PATH'], 'w')
32
33     q = int(input().strip())
34
35     for q_itr in range(q):
36         s = input()
37
38         result = anagram(s)
39
40         fptr.write(str(result) + '\n')
41
42     fptr.close()
43

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Hidden case	 Success	5	0.0428 sec	9.1 KB
Testcase 2	Easy	Hidden case	 Success	5	0.055 sec	9.19 KB
Testcase 3	Easy	Hidden case	 Success	5	0.0447 sec	9.31 KB
Testcase 4	Easy	Hidden case	 Success	5	0.0396 sec	9.17 KB
Testcase 5	Easy	Hidden case	 Success	5	0.0511 sec	9.15 KB
Testcase 6	Easy	Hidden case	 Success	5	0.1498 sec	9.25 KB
Testcase 7	Easy	Hidden case	 Success	5	0.0959 sec	9.36 KB
Testcase 8	Easy	Hidden case	 Success	5	0.0842 sec	9.35 KB
Testcase 9	Easy	Hidden case	 Success	5	0.0894 sec	9.33 KB
Testcase 10	Easy	Hidden case	 Success	5	0.1324 sec	9.49 KB
Testcase 11	Easy	Hidden case	 Success	5	0.0714 sec	9.32 KB
Testcase 12	Easy	Hidden case	 Success	5	0.0742 sec	9.24 KB
Testcase 13	Easy	Hidden case	 Success	5	0.0941 sec	9.21 KB
Testcase 14	Easy	Hidden case	 Success	5	0.0805 sec	9.23 KB

Testcase 15	Easy	Sample case		Success	0	0.0863 sec	9.29 KB
Testcase 16	Easy	Sample case		Success	0	0.0447 sec	9.25 KB

No Comments