



Full Name: pravallika

Email: pravallikajoseph20@gmail.com

Test Name: Mock Test

Taken On: 25 Aug 2025 09:46:12 IST

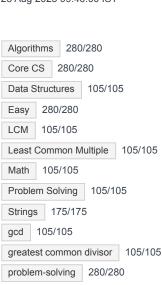
Time Taken: 84 min 4 sec/ 90 min

Invited by: Ankush

Invited on: 25 Aug 2025 09:46:00 IST

Skills Score:

Tags Score:



sets 105/105

100%

scored in **Mock Test** in 84 min 4 sec on 25 Aug 2025 09:46:12 IST

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review it in detail here -

Question Description	Time Taken	Score	Status
Q1 Palindrome Index > Coding	31 min 35 sec	105/ 105	(!)
Q2 Between Two Sets > Coding	19 min 34 sec	105/ 105	(!)
Q3 Anagram > Coding	31 min 4 sec	70/ 70	1

QUESTION 1

Score 105

Needs Review

Palindrome Index > 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 = "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

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

Sample Input

```
STDIN Function

-----

3  q = 3

aaab  s = 'aaab' (first query)

baa  s = 'baa' (second query)

aaa  s = '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 b' results in a palindrome, so return b'.

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.

Language used: C

```
1 int isPalindrome(char *s, int 1, int r) {
    while(l < r) {
        if(s[l] != s[r]){
4
            return 0;
         }
         1++;
         r--;
8
     }
     return 1;
10 }
11 int palindromeIndex(char* s) {
     int n = strlen(s);
     int 1 = 0, r = n-1;
     while(l<r){
14
     if(s[l] != s[r]){
            if(isPalindrome(s,l+1,r)){
                 return 1;
            if(isPalindrome(s,1,r-1)){
                return r;
             return -1;
         }
         1++;
         r--;
     }
      return -1;
28 }
```

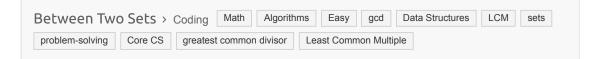
TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.008 sec	7.13 KB
Testcase 2	Medium	Hidden case	Success	5	0.0079 sec	7 KB
Testcase 3	Medium	Hidden case	Success	5	0.0067 sec	7 KB
Testcase 4	Medium	Hidden case	Success	5	0.0082 sec	6.88 KB
Testcase 5	Medium	Hidden case	Success	5	0.0066 sec	7.25 KB
Testcase 6	Medium	Hidden case	Success	5	0.0096 sec	7.38 KB
Testcase 7	Medium	Hidden case	Success	5	0.009 sec	7.13 KB
Testcase 8	Medium	Hidden case	Success	5	0.0086 sec	7.75 KB
Testcase 9	Hard	Hidden case	Success	10	0.0093 sec	7.5 KB
Testcase 10	Hard	Hidden case	Success	10	0.0073 sec	7.13 KB
Testcase 11	Hard	Hidden case	Success	10	0.0074 sec	7.13 KB
Testcase 12	Hard	Hidden case	Success	10	0.0112 sec	6.88 KB
Testcase 13	Hard	Hidden case	Success	10	0.0109 sec	7.25 KB
Testcase 14	Hard	Hidden case	Success	10	0.01 sec	7.38 KB
Testcase 15	Hard	Hidden case	Success	10	0.0082 sec	7.38 KB

QUESTION DESCRIPTION



Needs Review

Score 105



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 $0 \le i < n$.

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

Constraints

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

Sample Input

2 4

16 32 96

Sample Output

-

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
```

```
1 int gcd(int a, int b) {
2 while(b != 0){
         int temp = b;
4
        b = a%b;
         a =temp;
     }
      return a;
8 }
9 int lcm(int a, int b) {
     return (a*b)/gcd(a,b);
11 }
12 int getTotalX(int a_count, int* a, int b_count, int* b) {
   int l = a[0];
     for(int i = 1; i<a_count; i++){
14
          1 = lcm(l,a[i]);
     }
     int g = b[0];
     for(int i = 1; i < b_count; i++) {
       g = gcd(g,b[i]);
     }
     int count = 0;
     for (int x=1; x \le g; x += 1) {
     if(g % x == 0) {
             count++;
         }
      return count;
28 }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	Success	0	0.0101 sec	7.25 KB
Testcase 2	Easy	Hidden case	Success	15	0.0067 sec	7.38 KB
Testcase 3	Easy	Hidden case	Success	15	0.0069 sec	7.13 KB
Testcase 4	Easy	Hidden case	Success	15	0.0113 sec	7 KB
Testcase 5	Easy	Hidden case	Success	15	0.0074 sec	6.88 KB
Testcase 6	Easy	Hidden case	Success	15	0.0114 sec	7 KB
Testcase 7	Easy	Hidden case	Success	15	0.0074 sec	6.88 KB
Testcase 8	Easy	Hidden case	Success	15	0.0071 sec	7.25 KB
Testcase 9	Easy	Sample case	Success	0	0.0065 sec	7 KB

No Comments



Anagram > 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 8.

Constraints

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

Sample Input

```
6
aaabbb
ab
abc
mnop
xyyx
xaxbbbxx
```

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.

CANDIDATE ANSWER

Language used: C

```
1 int anagram(char* s) {
      int len = strlen(s);
      if( len % 2 != 0){
 4
          return -1;
      }
      int half = len/2;
 6
      int freq[26]={0};
     for(int i = 0; i<half;i++){
 8
          freq[s[i]-'a']++;
      for(int i = half;i<len;i++) {</pre>
          freq[s[i]-'a']--;
      int changes = 0;
      for(int i =0; i<26; i++){
          if(freq[i]>0){
              changes += freq[i];
      }
      return changes;
21 }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Hidden case	Success	5	0.0071 sec	6.88 KB
Testcase 2	Easy	Hidden case	Success	5	0.0078 sec	7 KB
Testcase 3	Easy	Hidden case	Success	5	0.0077 sec	7 KB
Testcase 4	Easy	Hidden case	Success	5	0.0085 sec	7 KB
Testcase 5	Easy	Hidden case	Success	5	0.0096 sec	7 KB
Testcase 6	Easy	Hidden case	Success	5	0.0111 sec	7.75 KB
Testcase 7	Easy	Hidden case	Success	5	0.0136 sec	7.88 KB
Testcase 8	Easy	Hidden case	Success	5	0.0175 sec	7.88 KB
Testcase 9	Easy	Hidden case	Success	5	0.0143 sec	7.38 KB
Testcase 10	Easy	Hidden case	Success	5	0.0237 sec	7.5 KB
Testcase 11	Easy	Hidden case	Success	5	0.0134 sec	7.75 KB
Testcase 12	Easy	Hidden case	Success	5	0.0181 sec	7.75 KB
Testcase 13	Easy	Hidden case	Success	5	0.0097 sec	7.75 KB
Testcase 14	Easy	Hidden case	Success	5	0.0101 sec	7.75 KB
Testcase 15	Easy	Sample case	Success	0	0.008 sec	7 KB
Testcase 16	Easy	Sample case	Success	0	0.0122 sec	7 KB

No Comments