



Full Name:

Prasane Pantala

Email:

prasanepantala7@gmail.com

Test Name:

Mock Test

Taken On:

20 Aug 2025 16:53:17 IST

Time Taken:

45 min 3 sec/ 90 min

Invited by:

Ankush

Invited on:

20 Aug 2025 16:53:07 IST

Skills Score:

Tags Score:

100%

280/280

scored in **Mock Test** in 45 min 3 sec on 20 Aug 2025 16:53:17 IST

- Algorithms280/280
- Core CS280/280
- Data Structures105/105
- Easy280/280
- LCM105/105
- Least Common Multiple105/105
- Math105/105
- Problem Solving105/105
- Strings175/175
- gcd105/105
- greatest common divisor105/105
- problem-solving280/280
- sets105/105

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	11 min 39 sec	105/ 105	✔
Q2	Between Two Sets > Coding	16 min 2 sec	105/ 105	⚠
Q3	Anagram > Coding	9 min 42 sec	70/ 70	✔

QUESTION 1



Correct Answer

Score 105

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

 $s = \text{"bcbc"}$

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: C

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int palindromeIndex(char* s) {
5     int l = 0, r = strlen(s) - 1;
6     while (l < r && s[l] == s[r]) {
7         l++;
8         r--;
9     }
10    if (l >= r) return -1;
11    int ll = l + 1, rr = r;
12    int flag1 = 1, flag2 = 1;
13    while (ll < rr) {
14        if (s[ll] != s[rr]) {
15            flag1 = 0;
16            break;
17        }
18        ll++; rr--;
19    }
20    ll = l; rr = r - 1;
21    while (ll < rr) {
22        if (s[ll] != s[rr]) {
23            flag2 = 0;
24            break;
25        }
26        ll++; rr--;
27    }
28    if (flag1) return l;
29    if (flag2) return r;
30    return -1;
31 }
32
33 int main() {
34     int q;
35     scanf("%d", &q);
36     char s[100005];
37     for (int i = 0; i < q; i++) {
38         scanf("%s", s);
39         int result = palindromeIndex(s);
40         printf("%d\n", result);
41     }
42     return 0;
43 }
44

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✔ Success	0	0.0086 sec	7.38 KB
Testcase 2	Medium	Hidden case	✔ Success	5	0.0073 sec	6.88 KB
Testcase 3	Medium	Hidden case	✔ Success	5	0.0071 sec	7.13 KB
Testcase 4	Medium	Hidden case	✔ Success	5	0.0071 sec	7.13 KB
Testcase 5	Medium	Hidden case	✔ Success	5	0.0067 sec	7.38 KB
Testcase 6	Medium	Hidden case	✔ Success	5	0.008 sec	7.25 KB
Testcase 7	Medium	Hidden case	✔ Success	5	0.0078 sec	7.13 KB

Testcase 8	Medium	Hidden case	✔ Success	5	0.01 sec	7.5 KB
Testcase 9	Hard	Hidden case	✔ Success	10	0.0084 sec	7.13 KB
Testcase 10	Hard	Hidden case	✔ Success	10	0.0081 sec	7.25 KB
Testcase 11	Hard	Hidden case	✔ Success	10	0.009 sec	7.38 KB
Testcase 12	Hard	Hidden case	✔ Success	10	0.0073 sec	7.13 KB
Testcase 13	Hard	Hidden case	✔ Success	10	0.008 sec	7.38 KB
Testcase 14	Hard	Hidden case	✔ Success	10	0.0096 sec	7.13 KB
Testcase 15	Hard	Hidden case	✔ Success	10	0.0087 sec	7.25 KB

No Comments

QUESTION 2



Needs Review

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 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: C

```
1  #include <stdio.h>
2
3  int computeGCD(int number1, int number2) {
4      while (number2 != 0) {
5          int temp = number2;
6          number2 = number1 % number2;
7          number1 = temp;
8      }
9      return number1;
10 }
11
12 int computeLCM(int number1, int number2) {
13     return (number1 / computeGCD(number1, number2)) * number2;
14 }
15
16 int countValidIntegers(int sizeA, int* setA, int sizeB, int* setB) {
17     int lcmOfSetA = setA[0];
18     for (int i = 1; i < sizeA; i++) {
19         lcmOfSetA = computeLCM(lcmOfSetA, setA[i]);
20     }
21
22     int gcdOfSetB = setB[0];
23     for (int i = 1; i < sizeB; i++) {
24         gcdOfSetB = computeGCD(gcdOfSetB, setB[i]);
25     }
26
27     int validCount = 0;
28     for (int currentMultiple = lcmOfSetA; currentMultiple <= gcdOfSetB;
29     currentMultiple += lcmOfSetA) {
30         if (gcdOfSetB % currentMultiple == 0) {
31             validCount++;
32         }
33     }
34
35     return validCount;
36 }
37
38 int main() {
39     int arraySize1, arraySize2;
40     scanf("%d %d", &arraySize1, &arraySize2);
41
42     int array1[arraySize1], array2[arraySize2];
43
44     for (int i = 0; i < arraySize1; i++) {
```

```

45     scanf("%d", &array1[i]);
46 }
47
48 for (int i = 0; i < arraySize2; i++) {
49     scanf("%d", &array2[i]);
50 }
51
52 int finalResult = countValidIntegers(arraySize1, array1, arraySize2,
53 array2);
54 printf("%d\n", finalResult);
55
56 return 0;
57 }

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✔ Success	0	0.0088 sec	7.13 KB
Testcase 2	Easy	Hidden case	✔ Success	15	0.0075 sec	7.5 KB
Testcase 3	Easy	Hidden case	✔ Success	15	0.0078 sec	7.13 KB
Testcase 4	Easy	Hidden case	✔ Success	15	0.0072 sec	7 KB
Testcase 5	Easy	Hidden case	✔ Success	15	0.0073 sec	7.25 KB
Testcase 6	Easy	Hidden case	✔ Success	15	0.0107 sec	7.25 KB
Testcase 7	Easy	Hidden case	✔ Success	15	0.0081 sec	7.5 KB
Testcase 8	Easy	Hidden case	✔ Success	15	0.0073 sec	7.38 KB
Testcase 9	Easy	Sample case	✔ Success	0	0.0069 sec	7.13 KB

No Comments

QUESTION 3



Correct Answer

Score 70

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 *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
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.

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: C

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int anagramChange(char *s) {
5     int n = strlen(s);
6     if (n % 2 != 0) return -1;
7
8     int m = n / 2, lc[26] = {0}, rc[26] = {0};
9     for (int i = 0; i < m; i++) lc[s[i] - 'a']++;
10    for (int i = m; i < n; i++) rc[s[i] - 'a']++;
11
12    int res = 0;
13    for (int i = 0; i < 26; i++)
14        if (lc[i] > rc[i]) res += lc[i] - rc[i];
15
16    return res;
17 }
```

```

18
19 int main() {
20     int t;
21     scanf("%d", &t);
22     char str[10005];
23     while (t--) {
24         scanf("%s", str);
25         printf("%d\n", anagramChange(str));
26     }
27     return 0;
28 }
29

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Hidden case	✔ Success	5	0.0084 sec	7.38 KB
Testcase 2	Easy	Hidden case	✔ Success	5	0.0081 sec	7.38 KB
Testcase 3	Easy	Hidden case	✔ Success	5	0.008 sec	7.25 KB
Testcase 4	Easy	Hidden case	✔ Success	5	0.008 sec	7.38 KB
Testcase 5	Easy	Hidden case	✔ Success	5	0.0114 sec	7 KB
Testcase 6	Easy	Hidden case	✔ Success	5	0.0204 sec	7.5 KB
Testcase 7	Easy	Hidden case	✔ Success	5	0.0155 sec	7.25 KB
Testcase 8	Easy	Hidden case	✔ Success	5	0.0233 sec	7.38 KB
Testcase 9	Easy	Hidden case	✔ Success	5	0.0123 sec	7.25 KB
Testcase 10	Easy	Hidden case	✔ Success	5	0.0115 sec	7.25 KB
Testcase 11	Easy	Hidden case	✔ Success	5	0.0128 sec	7.25 KB
Testcase 12	Easy	Hidden case	✔ Success	5	0.0214 sec	7.38 KB
Testcase 13	Easy	Hidden case	✔ Success	5	0.0384 sec	7.5 KB
Testcase 14	Easy	Hidden case	✔ Success	5	0.0119 sec	7.13 KB
Testcase 15	Easy	Sample case	✔ Success	0	0.0073 sec	7.38 KB
Testcase 16	Easy	Sample case	✔ Success	0	0.0081 sec	7.38 KB

No Comments