Mock Test › mhbagheri3@gmail.com

Full Name:	Mohammad Hossein Bagheri
Email:	mhbagheri3@gmail.com
Test Name:	**Mock Test**
Taken On:	12 Oct 2023 13:48:38 IST
Time Taken:	37 min 51 sec/ 90 min
Linkedin:	http://www.linkedin.com/in/mhbagheri
Invited by:	Ankush
Invited on:	12 Oct 2023 13:48:12 IST
Skills Score:	
Tags Score:	

Algorithms	280/280
Core CS	280/280
Data Structures	105/105
Easy	280/280
LCM	105/105
Least Common Multiple	105/105
Math	105/105
Problem Solving	105/105
Strings	175/175
gcd	105/105
greatest common divisor	105/105
problem-solving	280/280
sets	105/105

100%

280/280

scored in **Mock Test** in 37 min 51 sec on 12 Oct 2023 13:48:38 IST

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Palindrome Index › **Coding**	15 min 11 sec	105/ 105	✓
Q2	Between Two Sets › **Coding**	18 min 20 sec	105/ 105	✓
Q3	Anagram › **Coding**	2 min 25 sec	70/ 70	✓

QUESTION 1

✓

Correct Answer

Palindrome Index › Coding

Strings | Algorithms | Easy | problem-solving | Core CS | Problem Solving

QUESTION DESCRIPTION

Score 105

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

- $1 \le q \le 20$
- $1 \le \text{length of } s \le 10^5 + 5$
- 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.

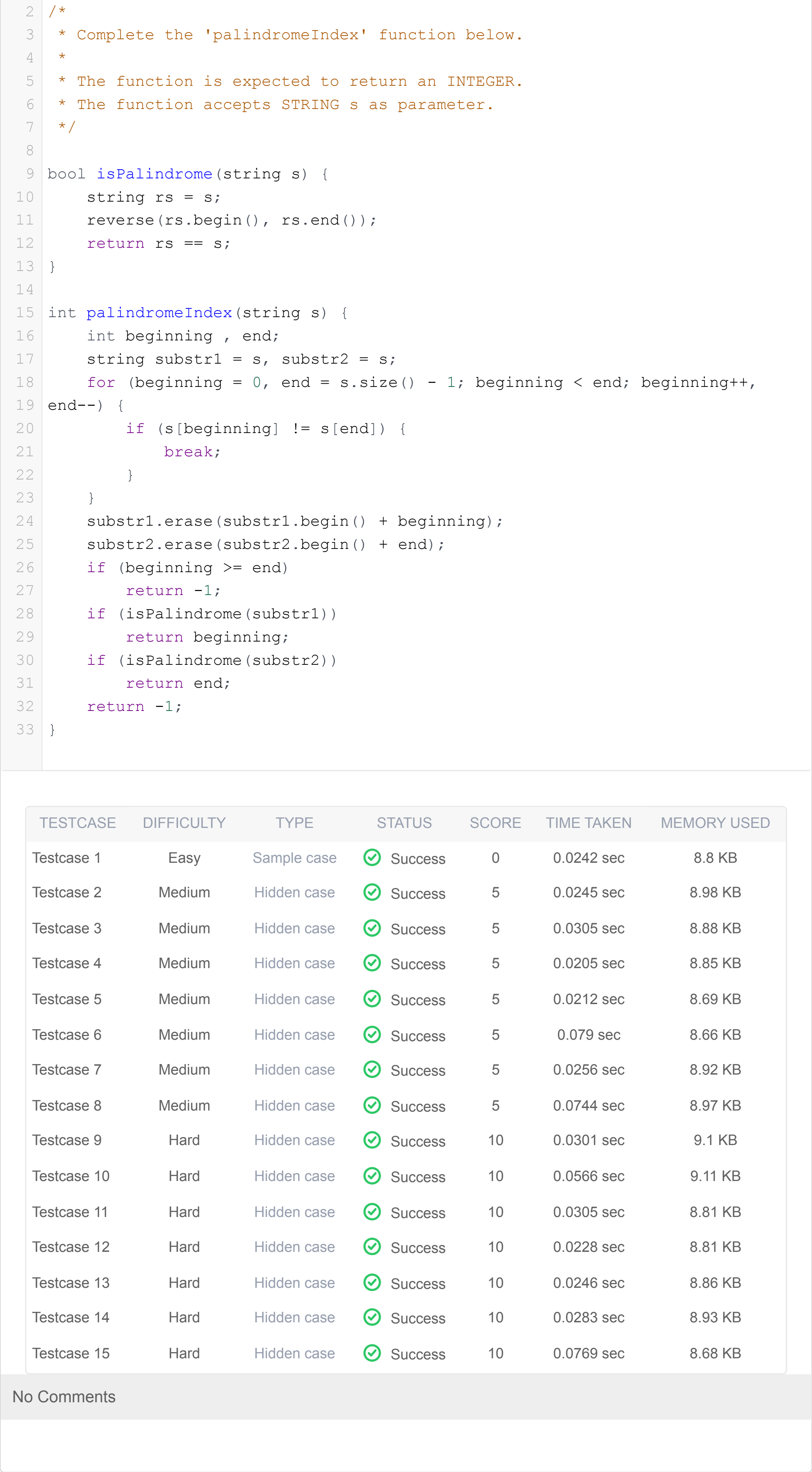
Query 2: "baa"
Removing *'b'* at index 0 results in a palindrome, so return 0.

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++14**



```cpp
/*
 * Complete the 'palindromeIndex' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts STRING s as parameter.
 */

bool isPalindrome(string s) {
    string rs = s;
    reverse(rs.begin(), rs.end());
    return rs == s;
}

int palindromeIndex(string s) {
    int beginning , end;
    string substr1 = s, substr2 = s;
    for (beginning = 0, end = s.size() - 1; beginning < end; beginning++,
end--) {
        if (s[beginning] != s[end]) {
            break;
        }
    }
    substr1.erase(substr1.begin() + beginning);
    substr2.erase(substr2.begin() + end);
    if (beginning >= end)
        return -1;
    if (isPalindrome(substr1))
        return beginning;
    if (isPalindrome(substr2))
        return end;
    return -1;
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✓ Success	0	0.0242 sec	8.8 KB
Testcase 2	Medium	Hidden case	✓ Success	5	0.0245 sec	8.98 KB
Testcase 3	Medium	Hidden case	✓ Success	5	0.0305 sec	8.88 KB
Testcase 4	Medium	Hidden case	✓ Success	5	0.0205 sec	8.85 KB
Testcase 5	Medium	Hidden case	✓ Success	5	0.0212 sec	8.69 KB
Testcase 6	Medium	Hidden case	✓ Success	5	0.079 sec	8.66 KB
Testcase 7	Medium	Hidden case	✓ Success	5	0.0256 sec	8.92 KB
Testcase 8	Medium	Hidden case	✓ Success	5	0.0744 sec	8.97 KB
Testcase 9	Hard	Hidden case	✓ Success	10	0.0301 sec	9.1 KB
Testcase 10	Hard	Hidden case	✓ Success	10	0.0566 sec	9.11 KB
Testcase 11	Hard	Hidden case	✓ Success	10	0.0305 sec	8.81 KB
Testcase 12	Hard	Hidden case	✓ Success	10	0.0228 sec	8.81 KB
Testcase 13	Hard	Hidden case	✓ Success	10	0.0246 sec	8.86 KB
Testcase 14	Hard	Hidden case	✓ Success	10	0.0283 sec	8.93 KB
Testcase 15	Hard	Hidden case	✓ Success	10	0.0769 sec	8.68 KB

No Comments

QUESTION 2

✓ 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**

```cpp
#include <bits/stdc++.h>
#include <algorithm>
using namespace std;

string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);



/*
 * Complete the 'getTotalX' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts following parameters:
 *  1. INTEGER_ARRAY a
 *  2. INTEGER_ARRAY b
 */

int gcd(int a, int b) {
  if (b == 0) {
    return a;
  } else {
    return gcd(b, a % b);
  }
}

int lcm(int a, int b) {
  return (a * b) / gcd(a, b);
}

int getTotalX(vector<int> a, vector<int> b) {
    int l = 1, g = 0;
    for(int i : a)
        l = lcm(l, i);

    for(int i : b)
        g = gcd(g, i);

    int cnt = 0;
    int num = l;
    while (num <= g) {
        if (g % num == 0) {
            cnt += 1;
        }
        num += l;
    }
    return cnt;
}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    string first_multiple_input_temp;
    getline(cin, first_multiple_input_temp);


    vector<string> first_multiple_input =
split(rtrim(first_multiple_input_temp));

    int n = stoi(first_multiple_input[0]);

    int m = stoi(first_multiple_input[1]);

    string arr_temp_temp;
    getline(cin, arr_temp_temp);

    vector<string> arr_temp = split(rtrim(arr_temp_temp));

    vector<int> arr(n);

    for (int i = 0; i < n; i++) {
        int arr_item = stoi(arr_temp[i]);

        arr[i] = arr_item;
    }

    string brr_temp_temp;
    getline(cin, brr_temp_temp);

    vector<string> brr_temp = split(rtrim(brr_temp_temp));

    vector<int> brr(m);

    for (int i = 0; i < m; i++) {
        int brr_item = stoi(brr_temp[i]);

        brr[i] = brr_item;
    }

    int total = getTotalX(arr, brr);

    fout << total << "\n";

    fout.close();

    return 0;
}

string ltrim(const string &str) {
    string s(str);

    s.erase(
        s.begin(),
        find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
    );

    return s;
}

string rtrim(const string &str) {
    string s(str);

    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>
(isspace))).base(),
        s.end()
    );

    return s;
```

```
}

vector<string> split(const string &str) {
    vector<string> tokens;

    string::size_type start = 0;
    string::size_type end = 0;

    while ((end = str.find(" ", start)) != string::npos) {
        tokens.push_back(str.substr(start, end - start));

        start = end + 1;
    }

    tokens.push_back(str.substr(start));

    return tokens;
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Sample case	✓ Success	0	0.0341 sec	8.99 KB
Testcase 2	Easy	Hidden case	✓ Success	15	0.0375 sec	8.83 KB
Testcase 3	Easy	Hidden case	✓ Success	15	0.0284 sec	8.95 KB
Testcase 4	Easy	Hidden case	✓ Success	15	0.0701 sec	8.89 KB
Testcase 5	Easy	Hidden case	✓ Success	15	0.0697 sec	8.8 KB
Testcase 6	Easy	Hidden case	✓ Success	15	0.0301 sec	8.86 KB
Testcase 7	Easy	Hidden case	✓ Success	15	0.0333 sec	8.65 KB
Testcase 8	Easy	Hidden case	✓ Success	15	0.0258 sec	8.82 KB
Testcase 9	Easy	Sample case	✓ Success	0	0.031 sec	8.73 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

$s = \text{abccde}$

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++14**

```

/*
 * Complete the 'anagram' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts STRING s as parameter.
 */

int anagram(string s) {
    map<char, int> m;
    int result = 0;

    if(s.size() % 2 == 1)
        return -1;

    for(int i = 0; i < s.size() / 2; i++)
        m[s[i]]++;

    for(int i = s.size() / 2; i < s.size(); i++){
        if(m[s[i]] != 0)
            m[s[i]]--;
        else
            result++;
    }
    return result;
}

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Hidden case	✓ Success	5	0.0301 sec	8.93 KB
Testcase 2	Easy	Hidden case	✓ Success	5	0.084 sec	8.77 KB
Testcase 3	Easy	Hidden case	✓ Success	5	0.0394 sec	8.82 KB
Testcase 4	Easy	Hidden case	✓ Success	5	0.0437 sec	8.78 KB
Testcase 5	Easy	Hidden case	✓ Success	5	0.0399 sec	8.8 KB
Testcase 6	Easy	Hidden case	✓ Success	5	0.0722 sec	8.88 KB
Testcase 7	Easy	Hidden case	✓ Success	5	0.0404 sec	8.84 KB
Testcase 8	Easy	Hidden case	✓ Success	5	0.0613 sec	8.84 KB
Testcase 9	Easy	Hidden case	✓ Success	5	0.0411 sec	8.85 KB
Testcase 10	Easy	Hidden case	✓ Success	5	0.1003 sec	8.79 KB
Testcase 11	Easy	Hidden case	✓ Success	5	0.0417 sec	9.02 KB
Testcase 12	Easy	Hidden case	✓ Success	5	0.0831 sec	8.91 KB
Testcase 13	Easy	Hidden case	✓ Success	5	0.0767 sec	8.84 KB
Testcase 14	Easy	Hidden case	✓ Success	5	0.0606 sec	8.79 KB
Testcase 15	Easy	Sample case	✓ Success	0	0.0674 sec	8.72 KB
Testcase 16	Easy	Sample case	✓ Success	0	0.027 sec	8.68 KB

No Comments

PDF generated at: 12 Oct 2023 08:58:15 UTC