

Mock Test > dzabeligan@gmail.com

Full Name:

Elijah Balogun

Email:

dzabeligan@gmail.com

Test Name:

**Mock Test** 

Taken On:

28 Jul 2023 22:22:43 IST

Time Taken:

11 min 50 sec/ 30 min

Invited by:

Ankush

Invited on:

28 Jul 2023 22:19:16 IST

Skills Score:

Tags Score:

Algorithms 70/70

Core CS 70/70

Easy 70/70

Strings 70/70

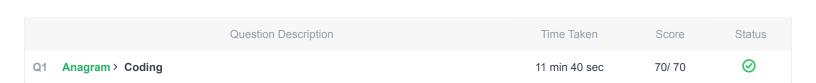
problem-solving 70/70

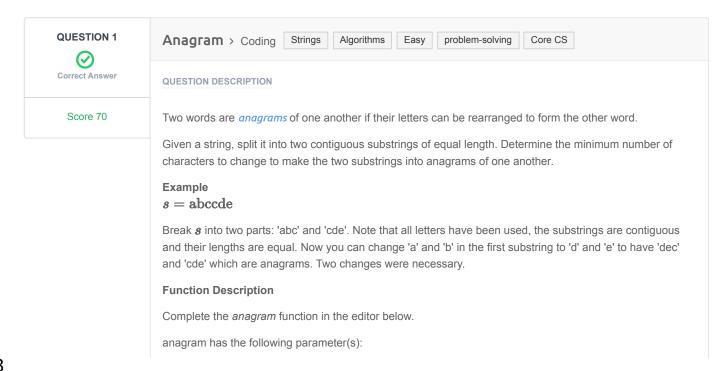
100% 70/70

scored in **Mock Test** in 11 min 50 sec on 28 Jul 2023 22:22:43 IST

### **Recruiter/Team Comments:**

No Comments.





• 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**

- $1 \le q \le 100$
- $1 \le |s| \le 10^4$
- ${\it s}$  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: Python 3

```
#
2 # Complete the 'anagram' function below.
3 #
4 # The function is expected to return an INTEGER.
5 # The function accepts STRING s as parameter.
6 #
7
8 def anagram(s):
9 # Write your code here
```

```
if len(s) & 1:
          return -1
      s1_count = {}
       s2_count = {}
14
      for letter in s[:int(len(s)/2)]:
           s1\_count[letter] = s1\_count.get(letter, 0) + 1
      for letter in s[int(len(s)/2):]:
           s2\_count[letter] = s2\_count.get(letter, 0) + 1
      counter = 0
      for key, value in s1_count.items():
           if key in s2_count:
              counter += abs(value - s2_count[key])
              s2_count.pop(key)
          else:
              counter += value
       for value in s2_count.values():
           counter += value
      return counter//2
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Hidden case	Success	5	0.0784 sec	10.7 KB
Testcase 2	Easy	Hidden case	Success	5	0.0689 sec	10.6 KB
Testcase 3	Easy	Hidden case	Success	5	0.1514 sec	10.8 KB
Testcase 4	Easy	Hidden case	Success	5	0.0771 sec	10.7 KB
Testcase 5	Easy	Hidden case	Success	5	0.1579 sec	10.8 KB
Testcase 6	Easy	Hidden case	Success	5	0.1459 sec	10.9 KB
Testcase 7	Easy	Hidden case	Success	5	0.0781 sec	10.9 KB
Testcase 8	Easy	Hidden case	Success	5	0.1756 sec	10.8 KB
Testcase 9	Easy	Hidden case	Success	5	0.1717 sec	10.8 KB
Testcase 10	Easy	Hidden case	Success	5	0.1187 sec	10.7 KB
Testcase 11	Easy	Hidden case	Success	5	0.075 sec	10.7 KB
Testcase 12	Easy	Hidden case	Success	5	0.1398 sec	10.8 KB
Testcase 13	Easy	Hidden case	Success	5	0.0944 sec	10.9 KB
Testcase 14	Easy	Hidden case	Success	5	0.1197 sec	10.8 KB
Testcase 15	Easy	Sample case	Success	0	0.1132 sec	10.7 KB
Testcase 16	Easy	Sample case	Success	0	0.1127 sec	10.6 KB

No Comments

PDF generated at: 28 Jul 2023 17:06:24 UTC