H

Full Name: Remi Chartier

Email: remipr.chartier@gmail.com

Test Name: Mock Test

Taken On: 26 Nov 2021 04:28:31 IST

Time Taken: 3 min 45 sec/ 30 min

Contact Number: +14084751573

Linkedin: http://www.linkedin.com/in/remichartier

Invited by: Ankush

Invited on: 26 Nov 2021 04:13:24 IST

Skills Score:

Tags Score: Algorithms 70/70

Core CS 70/70

Easy 70/70

Strings 70/70

problem-solving 70/70

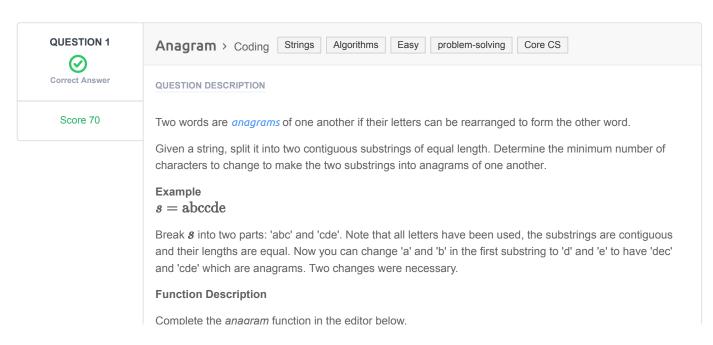


scored in **Mock Test** in 3 min 45 sec on 26 Nov 2021 04:28:31 IST

Recruiter/Team Comments:

No Comments.





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

- $1 \le q \le 100$
- $1 \le |s| \le 10^4$
- 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.

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 #
2 # Complete the 'anagram' function below.
3 #
4 # The function is expected to return an INTEGER.
5 # The function accepts STRING s as parameter.
#
```

```
8 def anagram(s):
9
     # Write your code here
      count = 0
     if len(s) % 2 != 0:
      return -1
     half = int(len(s)/2)
     # split string into 2 lists of equal length
14
     s1 = list(s[0:half])
     s2 = list(s[half:])
     count_s1 = {}
     for c in s1:
         if count s1.get(c) == None:
              count_s1[c] = 1
         else:
             count_s1[c] += 1
     for c in count_s1.keys():
          if count_s1[c] > s2.count(c):
              count += count_s1[c] - s2.count(c);
      return count
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Hidden case		5	0.0592 sec	9.47 KB
Testcase 2	Easy	Hidden case	Success	5	0.0447 sec	9.46 KB
Testcase 3	Easy	Hidden case	Success	5	0.0584 sec	9.39 KB
Testcase 4	Easy	Hidden case	Success	5	0.0429 sec	9.48 KB
Testcase 5	Easy	Hidden case	Success	5	0.0519 sec	9.42 KB
Testcase 6	Easy	Hidden case	Success	5	0.5017 sec	9.52 KB
Testcase 7	Easy	Hidden case	Success	5	0.1653 sec	9.59 KB
Testcase 8	Easy	Hidden case	Success	5	0.4947 sec	9.61 KB
Testcase 9	Easy	Hidden case	Success	5	0.1538 sec	9.6 KB
Testcase 10	Easy	Hidden case	Success	5	0.4688 sec	9.62 KB
Testcase 11	Easy	Hidden case	Success	5	0.1682 sec	9.54 KB
Testcase 12	Easy	Hidden case	Success	5	0.4927 sec	9.53 KB
Testcase 13	Easy	Hidden case	Success	5	0.4787 sec	9.44 KB
Testcase 14	Easy	Hidden case	Success	5	0.4674 sec	9.59 KB
Testcase 15	Easy	Sample case	Success	0	0.0506 sec	9.4 KB
Testcase 16	Easy	Sample case		0	0.0442 sec	9.3 KB

No Comments

PDF generated at: 25 Nov 2021 23:03:46 UTC