Ĥ

Full Name:

Đào Nguyễn Huy Hoàng

Email:

hoang.dnh210380@sis.hust.edu.vn

Test Name:

Mock Test

Taken On:

9 Apr 2023 12:51:37 IST

Time Taken:

20 min 22 sec/ 30 min

Invited by:

Ankush

Invited on:

9 Apr 2023 12:51:23 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 20 min 22 sec on 9 Apr 2023 12:51:37

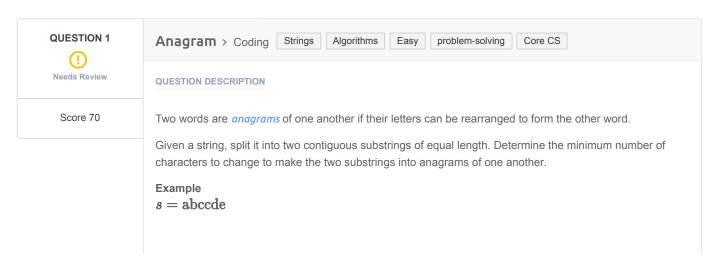
Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review.





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, \emph{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: C

1 #include <math.h>

```
2 #include <stdio.h>
3 #include <string.h>
4 #include <stdlib.h>
5 #include <assert.h>
6 #include <limits.h>
7 #include <stdbool.h>
8 int anagram(char* s){
      // Complete this function
      int l=strlen(s),c=0;
     if(1%2 != 0)
          return -1;
      int h1[26]={0}, h2[26]={0}, i;
14
      for(i=0;i<1/2;i++)
          h1[s[i]-'a']++;
      for(i=1/2;i<1;i++)
          h2[s[i]-'a']++;
      for(i=0;i<26;i++)
     {
          h1[i]=h1[i]-h2[i];
          if(h1[i]>0)
             c=c+abs(h1[i]);
24
      return c;
26 }
27 int main() {
      int q;
      scanf("%i", &q);
      for (int a0 = 0; a0 < q; a0++) {
          char* s = (char *) malloc(512000 * sizeof(char));
          scanf("%s", s);
          int result = anagram(s);
         printf("%d\n", result);
      return 0;
37 }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Hidden case	Success	5	0.0505 sec	7.42 KB
Testcase 2	Easy	Hidden case	Success	5	0.0547 sec	7.39 KB
Testcase 3	Easy	Hidden case	Success	5	0.0228 sec	7.46 KB
Testcase 4	Easy	Hidden case	Success	5	0.0221 sec	7.54 KB
Testcase 5	Easy	Hidden case	Success	5	0.057 sec	7.71 KB
Testcase 6	Easy	Hidden case	Success	5	0.0365 sec	8.5 KB
Testcase 7	Easy	Hidden case	Success	5	0.0586 sec	8.13 KB
Testcase 8	Easy	Hidden case	Success	5	0.0544 sec	8.47 KB
Testcase 9	Easy	Hidden case	Success	5	0.0711 sec	8.2 KB
Testcase 10	Easy	Hidden case	Success	5	0.0558 sec	8.59 KB
Testcase 11	Easy	Hidden case	Success	5	0.0561 sec	8.08 KB
Testcase 12	Easy	Hidden case	Success	5	0.0398 sec	8.62 KB
Testcase 13	Easy	Hidden case	Success	5	0.0337 sec	8.46 KB
Testcase 14	Easy	Hidden case	Success	5	0.0363 sec	8.67 KB
Testcase 15	Easy	Sample case	Success	0	0.0198 sec	7.55 KB
Testcase 16	Easy	Sample case	Success	0	0.0568 sec	7.4 KB

PDF generated at: 9 Apr 2023 07:43:37 UTC