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

Break 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, , the number of test cases.

Each test case will contain a string .

Constraints

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 into two strings =’aaa’ and =’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: and 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”.

Submissions: 166

Max Score: 25

Difficulty: Easy

Rate This Challenge:

More

1

#!/bin/python3

2

3

Import math

4

Import os

5

Import random

6

Import re

7

Import sys

8

9

#

10

# Complete the ‘anagram’ function below.

11

#

12

# The function is expected to return an INTEGER.

13

# The function accepts STRING s as parameter.

14

#

15

16

Def anagram(s):

17

# Write your code here

18

Strlen = len(s)

19

20

If strlen == 0 or strlen%2 != 0:

21

22

Return -1

23

24

Tochange = 0

25

26

Sublen = strlen//2

27

28

Lhs, rhs = s[:sublen], list(s[sublen:])

29

30

For char in lhs:

31

32

If char not in rhs:

33

34

Tochange += 1

35

36

Else:

37

38

Rhs.remove(char)

39

40

Return tochange

41

42

If \_\_name\_\_ == ‘\_\_main\_\_’:

43

Fptr = open(os.environ[‘OUTPUT\_PATH’], ‘w’)

44

45

Q = int(input().strip())

46

47

For q\_itr in range(q):

48

S = input()

49

50

Result = anagram(s)

51

52

Fptr.write(str(result) + ‘\n’)

53

54

Fptr.close()

55

Line: 40 Col: 20

Run Code Submit CodeUpload Code as File

Test against custom input

Testcase 0

Testcase 1

Congratulations, you passed the sample test case.

Click the Submit Code button to run your code against all the test cases.

Input (stdin)

6

Aaabbb

Ab

Abc

Mnop

Xyyx

Xaxbbbxx

Your Output (stdout)

3

1

-1

2

0

1

Expected Output

3

1

-1

2

0

1