Anagram

Two words are *anagrams* of one another if their letters can be rearranged to form the other word.

In this challenge, you will be given a string. You must split it into two contiguous substrings, then determine the minimum number of characters to change to make the two substrings into anagrams of one another.

For example, given the string 'abccde', you would break it 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.

Input Format

The first line will contain an integer, q, the number of test cases.
Each test case will contain a string s which will be concatenation of both the strings described above in the problem.
The given string will contain only characters in the range ascii[a-z].

Constraints

- $1 \leq q \leq 100$
- $1 \leq |s| \leq 10^4$
- s consists only of characters in the range ascii[a-z].

Output Format

For each test case, print an integer representing the minimum number of changes required to make an anagram. Print -1 if it is not possible.

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"*.