

Problem Statement

Sid is obsessed with reading short stories. Being a CS student, he is doing some interesting frequency analysis with the books. He chooses strings $S1$ and $S2$ in such a way that $|\text{len}(S1) - \text{len}(S2)| \leq 1$.

Your task is to help him find the minimum number of characters of the first string he needs to change to enable him to make it an [anagram](#) of the second string.

Note: A word x is an anagram of another word y if we can produce y by rearranging the letters of x .

Input Format

The first line will contain an integer, T , representing the number of test cases. Each test case will contain a string having length $\text{len}(S1) + \text{len}(S2)$, which will be concatenation of both the strings described above in the problem. The given string will contain only characters from a to z .

Output Format

An integer corresponding to each test case is printed in a different line, i.e. the number of changes required for each test case. Print -1 if it is not possible.

Constraints

- $1 \leq T \leq 100$
- $1 \leq \text{len}(S1) + \text{len}(S2) \leq 10^4$

Sample Input

```
6
aaabbb
ab
abc
mnop
xyyx
xaxbbbx
```

Sample Output

```
3
1
-1
2
0
1
```

Explanation

Test Case #01: We have to replace all three characters from the first string to make both of strings anagram. Here, $S1 = "aaa"$ and $S2 = "bbb"$. So the solution is to replace all character 'a' in string a with character 'b'.

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 anagram for each other.

Test Case #04: We have to replace both the characters of first string ("mn") to make it anagram of other

one.

Test Case #05: $S1$ and $S2$ are already anagram to each other.

Test Case #06: Here $S1 = "xaxb"$ and $S2 = "bbxx"$. He had to replace 'a' from $S1$ with 'b' so that $S1 = "xbxb"$ and we can rearrange its letter to "bbxx" in order to get $S2$.