Problem Statement

Suggest Edit

You have been given two strings, let's say "str1" and "str2" of equal lengths. You are supposed to return the minimum number of manipulations required to make the two strings anagrams.

An anagram is a word or phrase formed by rearranging the letters of a different word or phrase. We can generalise this in string processing by saying that an anagram of a string is another string with the same quantity of each character in it, in any order.

For Example:

```
String “eat” and “ate” are anagram to each other but string “buy” and “bye” are
not.
```

Input Format:

```
The first line contains an integer 'T', which denotes the number of test cases or
queries to be run. Then the test cases follow.

The first-line and second line of each test case contain the string "str1" and
"str2" corresponding.
```

Output Format:

```
For each test case/query, print the minimum number of manipulations required to
make "str1" and "str2" string anagram.

Output for every test case will be printed in a separate line. You do not need to
print anything; it has already been taken care of.
```

Constraints :

```
1<= T <= 100
1<= N <= 5*10^3

Where 'T' is the number of test cases, 'N' is the length of "str1" and "str2" and
“str1” and “str2” will only contain lowercase Latin letters.

Time Limit: 1sec
```

Input 1:

```
2
except
accept
```

```
buy
bye
```

Output 1 :

```
2
1
```

Explanation Of Input 1 :

```
In the first test cases, we can change two character of "str1" i.e. {'e','x'} to
{'a','c'} or we can change two character of "str2" i.e. {'a','c'} to {'e','x'}, to
make string anagram. So the minimum number of manipulations to make "str1" and
"str2" to anagram string will be 2.

In the second test case, we can change one character of "str1" i.e. {'u'} to {'e'}
or we can change one character of "str2" i.e. {'e'} to {'u'}, to make string
anagram. So the minimum number of manipulations to make "str1" and "str2" to
anagram string will be 1.
```

Input 2:

```
3
hear
here
mail
male
ninja
ninja
```

Output 2 :

```
1
1
0
```

Explanation Of Input 2 :

```
In the first test case, we can change one character of "str1" i.e. {'a'} to {'e'}
or we can change one character of "str2" i.e. {'e'} to {'a'}, to make string
anagram. So the minimum number of manipulations to make "str1" and "str2" to
anagram string will be 1.

In the second test case, we can change one character of "str1" i.e. {'i'} to {'e'}
or we can change one character of "str2" i.e. {'e'} to {'i'}, to make string
anagram. So the minimum number of manipulations to make "str1" and "str2" to
```

anagram string will be 1.

In the third test case, Both strings are already anagram. So we do not need to do any manipulation. So the minimum number of manipulations to make "str1" and "str2" to anagram string will be 0.