HackerRank

Strings: Making Anagrams

A student is taking a cryptography class and has found *anagrams* to be very useful. Two strings are anagrams of each other if the first string's letters can be rearranged to form the second string. In other words, both strings must contain the same exact letters in the same exact frequency. For example, `bacdc` and `dcbac` are anagrams, but `bacdc` and `dcbad` are not.

The student decides on an encryption scheme that involves two large strings. The encryption is dependent on the minimum number of character deletions required to make the two strings anagrams. Determine this number.

Given two strings, a and b, that may or may not be of the same length, determine the minimum number of character deletions required to make a and b anagrams. Any characters can be deleted from either of the strings.

Example
$a = \text{'cde'}$
$b = \text{'dcf'}$

Delete e from a and f from b so that the remaining strings are cd and dc which are anagrams. This takes 2 character deletions.

Function Description

Complete the *makeAnagram* function in the editor below.

makeAnagram has the following parameter(s):

- *string a*: a string
- *string b*: another string

Returns

- *int:* the minimum total characters that must be deleted

Input Format

The first line contains a single string, a.
The second line contains a single string, b.

Constraints

- $1 \leq |a|, |b| \leq 10^4$
- The strings a and b consist of lowercase English alphabetic letters, ascii[a-z].

Sample Input

```
cde
abc
```

Sample Output

```
4
```

Explanation

Delete the following characters from the strings make them anagrams:

1. Remove `d` and `e` from `cde` to get `c`.
2. Remove `a` and `b` from `abc` to get `c`.

It takes 4 deletions to make both strings anagrams.