Kick Start 2021 - Round H

Transform the String

Problem

You are given a string $\mathbf{S}$ which denotes a padlock consisting of lower case English letters. You are also given a string $\mathbf{F}$ consisting of set of favorite lower case English letters. You are allowed to perform several operations on the padlock. In each operation, you can change one letter of the string to the one following it or preceding it in the alphabetical order. For example: for the letter `c`, you are allowed to change it to either `b` or `d` in an operation. The letters can be considered in a cyclic order, i.e., the preceding letter for letter `a` would be letter `z`. Similarly, the following letter for letter `z` would be letter `a`.

Your aim is to find the minimum number of operations that are required such that each letter in string $\mathbf{S}$ after applying the operations, is present in string $\mathbf{F}$.

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow.

Each test case consists of two lines.
The first line of each test case contains the string $\mathbf{S}$.
The second line of each test case contains the string $\mathbf{F}$.

Output

For each test case, output one line containing `Case #`x`:` y, where x is the test case number (starting from 1) and y is the minimum number of operations that are required such that each letter in string $\mathbf{S}$ after applying the operations, is one of the characters in string $\mathbf{F}$.

Limits

Memory limit: 1 GB.
$1 \leq \mathbf{T} \leq 100$.
$1 \leq$ the length of $\mathbf{S} \leq 10^5$.
$\mathbf{S}$ only consists of lower case English letters.
$\mathbf{F}$ only consists of distinct lower case English letters.
The letters in string $\mathbf{F}$ are lexicographically sorted.

Test Set 1

Time limit: 20 seconds.
The length of $\mathbf{F} = 1$.

Test Set 2

Time limit: 40 seconds.
$1 \leq$ the length of $\mathbf{F} \leq 26$.

Sample

Note: there are additional samples that are not run on submissions down below.

Sample Input	Sample Output
2 abcd a pppp p	Case #1: 6 Case #2: 0

In Sample Case #1, all the letters in string $\mathbf{S}$ should be converted to letter `a`. We can keep on changing the letters to its preceding letter till we reach the letter `a`. We do not need to change the first letter as it is already `a`. The second letter needs 1 operation to change it to `a`. The third letter needs 2 operations to change it to `a`. The fourth letter needs 3 operation to change it to `a`. Hence, we need a total of 6 operations to change string $\mathbf{S}$ such that all letters are changed to `a`.

In Sample Case #2, string $\mathbf{S}$ already contains only the favorite letter from string $\mathbf{F}$. Hence, we do not require any more operations.

Additional Sample - Test Set 2

The following additional sample fits the limits of Test Set 2. It will not be run against your submitted solutions.

Sample Input	Sample Output
3 pqrst ou abd abd aaaaaaaaaaaaaaab aceg	Case #1: 9 Case #2: 0 Case #3: 1

In Sample Case #1, all the letters in string $\mathbf{S}$ should be converted to either the letter `o` or the letter `u`. For the first and second letters it is optimal to change them to preceding letters till they are changed to letter `o`. The first letter would take 1 operation to change to letter `o`. The second letter would take 2 operations to change to letter `o`. For fourth and fifth letters it is optimal to change them to following letters till they are changed to letter `u`. The fourth letter would take 2 operations to change to letter `u`. The fifth letter would take 1 operation to change to letter `u`. We can change the third letter to either `o` or `u` as both of them would require 3 operations. Hence, we need a total of 9 operations to change string $\mathbf{S}$ such that all letters are changed to either `o` or `u`.

In Sample Case #2, string $\mathbf{S}$ already contains only the favorite letters from string $\mathbf{F}$. Hence, we do not require any more operations.

In Sample Case #3, we only need to change the last letter `b` to either `a` or `c`. Thus, we only need 1 operation.