

Elly has two strings of the same length **A** and **B**. Now she wants to transform **A** into an anagram of **B** by changing some of its letters.

Definition: We call two strings X and Y anagrams of each-other if we can re-arrange the letters in one of them to get the other.

The only operation the girl can make is cyclically increment one or more times some (possibly none or all) of the characters in **A**. Example increments are 'A' to 'B', 'K' to 'L' or 'Z' to 'A'.

For example, she can transform "ELLY" to "KRIS" character by character by shifting 'E' to 'K' (6 operations), 'L' to 'R' (again 6 operations), the second 'L' to 'I' (23 operations, going from 'Z' to 'A' on the 15-th operation), and finally 'Y' to 'S' (20 operations, again cyclically going from 'Z' to 'A' on the 2-nd operation). The total number of operations would be $6 + 6 + 23 + 20 = 55$. However, to make "ELLY" an anagram of "KRIS" it would be better to change it to "IRSK" with only 29 operations.

Task

You are given the strings **A** and **B**. Write a program **anagrams** which calculates the minimal number of operations needed to transform **A** into some other string **X**, such that **X** is an anagram of **B**.

Input

On the first line of the standard input will be given the string **A** and on the next one – **B**. Both strings will contain only uppercase letters of the English alphabet ('A'-'Z').

Output

On the standard output print one integer – the minimum number of increments.

Constraints

$1 \leq |A| = |B| \leq 50$ ($|S|$ is the length of string S)

Example

<i>Sample Input</i>	<i>Sample Output</i>
ELLY KRIS	29
TOPCODER TCOINDIA	57
AWIODJGWAMBAUWNMQEROAIQWYRZSVE PTT BNSIAELDALCGAWOPIWEQTYCNAZEKJXVYU	105