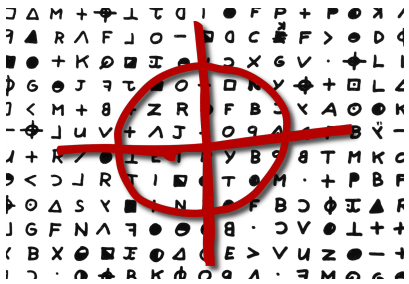
Problem D: Decrypting Zodiac



Original message from Zodiac

In the late 1960s, a serial killer committed his monstrous deeds. He was neither caught nor was he identified and due to a series of cryptic letters he sent to the press he was called *Zodiac*. It was assumed that those letters contain the killer's real name, but even to this day not all of them have been decrypted. One of the reasons for this is that the encrypted messages contain mistakes. It is not known if Zodiac made those mistakes on purpose to make the decryption harder.

For one of his first letters he used the following two step encryption scheme.[2] First, he applied a *Caesar cipher* which means that he replaced each letter with the one that comes k steps later in the alphabet, where k is a fixed number between 0 and 25 inclusive. Note that for this step it is assumed that after `z` the alphabet starts again with `a`. In the second step, he cut the message into two parts at an arbitrary position and swapped the parts. It is allowed for one of the two parts to be empty, in which case the message did not change during this second step.

Normally, a simple brute force search could be used to decrypt the message. However, to do this, one needs to automatically check if a message makes sense. Since Zodiac might have made some mistakes during the first step of the encryption, this is not easy to decide.

For this reason, you decided to try another approach. You want to try meaningful candidate sentences and encrypt them and then count how many mistakes would be required to make them match with Zodiac's encrypted message.

Input

The input consists of:

- One line with a single integer n ($1 \leq n \leq 1.5 \cdot 10^5$), the length of the messages.
- Two lines each with one string of length n. The first string is the encrypted message and the second string is your guess for the decrypted message.

Both strings consist of lowercase letters `a-z` only.

Output

Output a single integer, the minimal number of mistakes Zodiac must have made during the encryption, assuming you correctly guessed the decrypted message.

Sample Input 1	Sample Output 1
6 drhmex zodiac	2

[2] He did in fact not.

Sample Input 2	Sample Output 2
8 dicepara paradise	1

Sample Input 3	Sample Output 3
13 lvlvdvdqsonwk thisisasample	2

Notes

In the first sample the message can be encrypted by Caesar shifting each letter by four, resulting in `dshmeg`. After this, all letters match except for the second and sixth.

In the second example we can Caesar shift by zero, then split the message in the middle and swap both halves. After this, there is only a single mismatch: `s` $\leftrightarrow$ `c`.

In the third example the message can be encrypted by Caesar shifting by three in the first step, resulting in the message `wklvlvdvdpsoh`. Then, the first two letters can be cut off and swapped with the rest of the string to create `lvlvdvdpsohwk`. After this, only two letters will differ: `p` $\leftrightarrow$ `q` and `n` $\leftrightarrow$ `h`.