D. Two Strings Swaps

time limit per test
2 seconds
memory limit per test
256 megabytes
input
standard input
output
standard output

You are given two strings a and b consisting of lowercase English letters, both of length n. The characters of both strings have indices from 1 to n, inclusive.

You are allowed to do the following *changes*:

- Choose any index i ($1 \le i \le n$) and swap characters a_i and b_i;
- Choose any index i ($1 \le i \le n$) and swap characters a_i and a_{n-i+1};
- Choose any index i ($1 \le i \le n$) and swap characters b_i and b_{n-i+1}.

Note that if n is odd, you are formally allowed to swap $a_{\lceil \frac{n}{2} \rceil}$ with $a_{\lceil \frac{n}{2} \rceil}$ (and the same with the string b) but this move is useless. Also you can swap two equal characters but this operation is useless as well.

You have to make these strings equal by applying any number of *changes* described above, in any order. But it is obvious that it may be impossible to make two strings equal by these swaps.

In one *preprocess move* you can replace a character in a with another character. In other words, in a single *preprocess move* you can choose any index i ($1 \le i \le n$), any character c and set $a_i := c$.

Your task is to find the minimum number of *preprocess moves* to apply in such a way that after them you can make strings a and b equal by applying some number of *changes* described in the list above.

Note that the number of *changes* you make after the *preprocess moves* does not matter. Also note that you cannot apply *preprocess moves* to the string b or make any *preprocess moves* after the first *change* is made.

Input

The first line of the input contains one integer n ($1 \le n \le 10^5$) — the length of strings a and b.

The second line contains the string a consisting of exactly n lowercase English letters.

The third line contains the string b consisting of exactly n lowercase English letters.

Output

Print a single integer — the minimum number of *preprocess moves* to apply before *changes*, so that it is possible to make the string a equal to string b with a sequence of *changes* from the list above.

Examples

input

Copy

```
7
abacaba
bacabaa
```

output

Copy

```
4
```

input

Copy

```
5
zcabd
dbacz
```

output

Copy

```
0
```

Note

In the first example *preprocess moves* are as follows: $a_1:=$'b', $a_3:=$'c', $a_4:=$'a' and $a_5:=$'b'. Afterwards, $a=$"bbcabba". Then we can obtain equal strings by the following sequence of *changes*: $swap(a_2,b_2)$ and $swap(a_2,a_6)$. There is no way to use fewer than 4 *preprocess moves* before a sequence of *changes* to make string equal, so the answer in this example is 4.

In the second example no *preprocess moves* are required. We can use the following sequence of *changes* to make a and b equal: $swap(b_1,b_5)$, $swap(a_2,a_4)$.