問題 1 – 字串重排 (String Rearranging)

(10 分)

問題敍述

小明是一位很喜歡字串的少年，他最近在網路上發現一種計算兩個長度相同的字串的相似度的方法。該方法為：定義 s_1 為第一個字串，$s_1[i]$ 代表他的第 i 個字元，s_2 為第二個字串，$s_2[i]$ 代表他的第 i 個字元。兩個字串的相似度為有多少 i 使得 $s_1[i] = s_2[i]$。

舉例：aabbcc 與 azbbzc 的相似度為 4 ，因為 $s_1[1] = s_2[1]$ 、 $s_1[3] = s_2[3]$ 、 $s_1[4] = s_2[4]$ 、 $s_1[6] = s_2[6]$。

小明突發奇想，想測試看看若將第二個字串重新排列，那麼相似度最高能達到多少，但因為一個字串的排列數量非常多，小明無法測試全部的排列，因此希望你能寫一個程式幫助他。

輸入格式

第一行為一個整數 N 代表字串長度。
下一行輸入一個長度 N 且由 a 到 z 組成的字串 s_1。
最後一行輸入一個長度 N 且由 a 到 z 組成的字串 s_2。

輸出格式

請輸出在可以將 s_2 重新排列的情況下，兩個字串s_1, s_2 的所能達到的最高相似度為何。

資料範圍

- $1 \leq N \leq 1000$

輸入範例 1

```
5
abbcd
zbadp
```

輸出範例 1

```
3
```

輸入範例 2

```
4
yxxx
xxyx
```

輸出範例 2

```
4
```

輸入範例 3

```
3
ytp
gcc
```

輸出範例 3

```
0
```

範例說明

範例一，我們可以將第二個字串重新排列成為 abzpd，這樣他與 abbcd 的相似度為 3 。且不存在其他排列能讓相似度大於 3。

範例二，我們能將第二個字串重新排列後與第一個字串相等，因此相似度就是他們的長度也就是 4。

範例三，第二個字串的所有排列都不可能與第一個字串有大於 0 的相似度，因此輸出 0 。

Q1 - String Rearranging

(10 points)

Description

Ming is a youngster who is passionate about string. Recently, he discovered a method to determine the similarity of two given strings on the Internet. The definition of the method is: Define s_1 as the first string, and $s_1[i]$ represents the i-th character of s_1. Define s_2 as the second string, and $s_2[i]$ represents the i-th character of s_2. The similarity of s_1, s_2 is the number of i such that $s_1[i] = s_2[i]$.

For example, the similarity of aabbcc and azbbzc is 4 since $s_1[1] = s_2[1]$, $s_1[3] = s_2[3]$, $s_1[4] = s_2[4]$ and $s_1[6] = s_2[6]$.

Ming wants to check that what's the maximum similarity if he can rearrange the second string. Since the number of permutations of a string may be enormous, Ming isn't capable to check all the permutations. Please write a program to help Ming.

Input Format

The first line contains a single integer N representing the length of the string.
The next line contains a string s_1 whose length is N and consists of only a to z.
The last line contains a string s_2 whose length is N and consists of only a to z.

Output Format

Please output the maximum similarity of s_1, s_2 after rearranging s_2.

Data Range

- $1 \leq N \leq 1000$

Input Example 1

```
5
abbcd
zbadp
```

Output Example 1

```
3
```

Input Example 2

```
4
yxxx
xxyx
```

Output Example 2

```
4
```

Input Example 3

```
3
ytp
gcc
```

Output Example 3

```
0
```

Example Explanation:

Example 1: We can rearrange the second string to `abzpd` so that the similarity of `abzpd` and `abbcd` is 3, and there’s no permutation that can achieve higher similarity.

Example 2: We can rearrange the second string so that it’s identical to the first string. Therefore, the similarity is the length of the string which is 4.

Example 3: It’s impossible to rearrange second string to achieve the similarity that is greater than zero. So output 0.