

Perfect Cooking

Alex likes to cook delicious food and today he has decided to prepare Chicken Shawarma. Before getting started, he wrote down his recipe $X = x_0x_1x_2...x_{n-1}$ with n steps. To verify his steps, he referred to the recipe $Y = y_0y_1y_2...y_{m-1}$ with m steps, prepared by Ramsay Gordon himself. Now, Alex wants to make some updates in his recipe so that it is exactly the same as Gordon's recipe. In one update he can do the following in his own recipe, - remove a step, - add a step, - change a step.

Every step is represented by an uppercase letter of the English alphabet. Now, Alex wants to make minimum number of updates to his recipe to match Gordon's, after all he is too eager to start cooking. Find out the minimum number of updates he will need to make.

Input

Input contains two lines, such that the first line consists of the number of steps in the recipe X, that is n , and the recipe X itself as a string. n and X are separated by a space.

The second line consists of the number of steps in the recipe Y, that is m , and the recipe Y itself as a string. Again, m and Y are separated by a space. $0 < n, m \leq 100$.

Output

Output the minimum number of updates needed to convert X to Y. The output ends with a newline.

Example 1

Input :

5 ABCDE

4 ABDE

Output :

1

Example 2

Input :

3 TGZ

8 ACGTTGGC

Output :

7

Edit Distance

Edit distance is a way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other.

In this problem, we consider a single operation: replace a single character with another character.

For example, we can transform “stock” to “spice” with the following steps:

“stock” -> “stick” -> “slick” -> “slice” -> “spice”

Each successive word differs from the previous word in only a single character position while the word length remains the same. The added restriction is to use the words from a given dictionary in each step of the transformation.

Given a dictionary and a list of word-pairs, write a program to determine the minimum number of steps in the shortest possible transformation for each pair.

Input

The first line contains an integer **n**, the number of words in the dictionary, $1 \leq n \leq 200$. Each of the following **n** lines contains a word in the dictionary. All words will be alphabetic and in lower case. And no word will be longer than 10 characters.

The following line contains an integer **m**, the number of queries. Each of the following **m** lines contain a pair of word separated by space. They represent the starting word and the ending word. All pairs are guaranteed to have a transformation using the dictionary given.

Output **m** lines of numbers: the i-th line represents the minimum number of steps in the transformation for the i-th pair of words.

Example 1

Input:

```
11
dip
lip
mad
map
may
pad
pip
pod
pop
sap
sip
4
may mad
map map
pip pod
mad dip
```

Output:

```
1
0
2
4
```

Word play

Alan loves playing with words. He not only likes words that make sense but loves any sequence of alphanumeric characters. This time Alan is presented with two such sequences and he wants to find a subsequence which is common to both given sequences and has the maximum length possible.

Input

The input contains pairs of sequences. The first line of a pair is the first sequence and the second line contains the second sequence. The length of each sequence is at most 1000.

Output

For each pair of input lines, output a line containing one integer: the length of the maximal subsequence that both sequences have in common.

Example 1

Input :

```
bcacbcabbaccbab
bccabccbbabacbc
a1b2c3d4e
zz1yy2xx3ww4vv
abcdgh
aedfhr
abcdefghijklmnopqrstuvwxyz
a0b0c0d0e0f0g0h0i0j0k0l0m0n0o0p0q0r0s0t0u0v0w0x0y0z0
abcdefghijklmnopzyxwvutsrqpo
opqrstuvwxyzabcdefghijklmnopqrstuvwxyz
```

Output :

```
11
4
3
26
14
```

Toy Blocks

Ayu is playing with toy blocks. Ayu decides to build two towers with those blocks. She wants to use up all of the blocks she has and the number of blocks used in two towers should not differ by more than one. Besides, every block has a height and she wants to minimize the height difference between two towers.

Input

The first line of the input contains one integer N ($1 \leq N \leq 100$), the number of toy blocks. Each of the following N lines contains one integer indicating the height h ($1 \leq h \leq 450$) of that block.

Output

Print one line, containing two space separated integers, the heights of two towers. The smaller number goes first.

Example 1

Input :

3

100

90

200

Output :

190 200