Word sequence | 3 point(s)

Your friends are playing a game called "Word Sequence". In this game you are given a dictionary of words D and two words A and B(both words are in the dictionary), and you have to find the minimum number of changes needed to turn A into B.

A change is defined as replacing only one character from the current string with another, such that the new string is also in D.

Let's first consider D = [abc, abb, acc, ccc, bbb] and:

\* abc -> abb (this is a valid single change).

\* abc -> bbc -> bbb (this is NOT valid because bbc is NOT in `D`).

\* abc -> ccc (this is NOT a valid change because there are two replacements in one change).

Your task is to write a program that can find the minimum number of changes required for a given set of (A,B) pairs.

Input definition

The first line of the input gives the number of the words in the dictionary L (2 <= L <= 100)and the number of queries to solve Q (1 <= Q <= 10). The next L lines each contain a single word in the dictionary. The dictionary can contain words of varying length. The final Q lines of input each contain an A and B pair to evaluate.

It is guaranteed that all of the given words in the input only contain lower case characters and that the maximum length of any word is 10 characters.

Output definition

For each query in the input, your output should contain a single line that has the minimum number of changes required to turn string A into B. If there is no possible solution print "-1" instead (without the quotes).

To add clarity, let's now consider the specific input file given on this page.

* For the first query, the number of changes is 2 as you can transform 'abc' to 'acc' and then to 'ccc'.
* For the second query, you can transform acc to ccc in 1 change.
* For the third query, there is no way to transform abc to bbb; therefore, the output is -1.

Example input

5 3

abc

bbb

aab

acc

ccc

abc ccc

acc ccc

abc bbb

Example output

2

1

-1