7106 Word Ladder

A *Word Ladder* is a puzzle in which you transform one word into another, by changing one letter at a time. But, there's a catch: every word that you form in each step must be in the dictionary! Here's an example of how to transform `CAT` into `GAS`:

`CAT → CAR → WAR → WAS → GAS`

Of course, you want to use the fewest number of transitions possible. These puzzles can be tough, and often you'll think to yourself: "Darn it! If only [some word] was in the dictionary!"

Well, now is your chance! Given a dictionary, and a starting and ending word, what **ONE** single word could you add to the dictionary to minimize the number of steps to get from the starting word to the ending word, changing only one letter at a time, and making sure that every word at every step is in the dictionary?

Input

The input file contains several test cases, each of them as described below.

Each test case will start with a line with a single integer n ($2 \leq n \leq 1,000$) which indicates the number of words in the dictionary. The dictionary will follow on the next n lines, with one word per line. All words will consist of between 1 and 8 capital letters only, and all of the words in a test case will be of the same length. The first word in the list will be the starting word of the word ladder, and the second will be the ending word of the word ladder.

Output

For each test case, output exactly two lines. The first line holds the one single word that you would add to the dictionary, and the second holds an integer indicating the minimum number of steps to get from the starting word to the ending word, adding your word. Output no spaces.

It is possible that there's more than one word you can add that will make your path as short as possible. In this case, output the solution word that comes first alphabetically.

It is possible that there's no word you can add that will that will make your path any shorter. In this case, output 0 (zero) as the word.

It is possible that there's no word you can add that makes the solution possible. In this case, output '`0`' (zero) as the word, and '`-1`' as the number of steps.

Sample Input

```
3
CAT
DOG
COT
2
CAT
DOG
4
CAT
DOG
COT
```

```
COG
```

Sample Output

```
COG
3
0
-1
0
3
```