Problem K. Typo sequences

Source file name:	K.c, K.cpp, K.java, K.py
Input:	`Standard`
Output:	`Standard`

As part of a study about typos on modern smartphone keyboards, a volunteer is given a list of words and he is asked to type them all without correcting if they make a mistake. You've been hired to produce a program to analyze the results and determine which kinds of typos are more common.

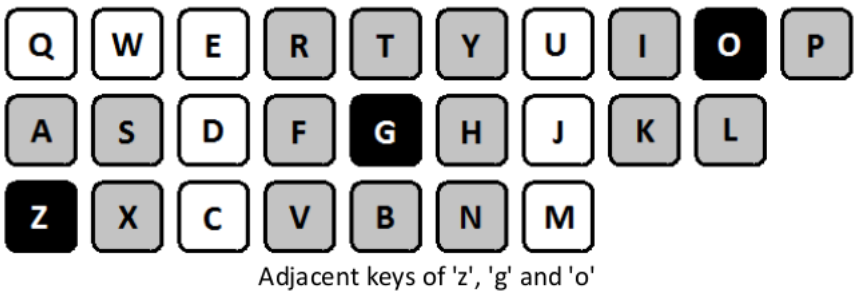
You're given a list of pairs of words, a reference word (the word the volunteer was supposed to type) and an output word (the word the volunteer actually typed).

For each pair of words, your job is to determine all the possible sequences of typos that the volunteer may have followed to produce the output word while attempting to type the reference word.

The volunteer attempted to type each letter of the reference word just once, and for each of those letters there are 5 possible outcomes when typing it:

1. OK (o): The volunteer typed the correct letter.
2. FORGOT (f): The volunteer forgot to type the letter.
   e.g. forgot to type an 'l' in 'hello', producing 'helo'.
3. WRONG (w): The volunteer typed an "adjacent" key instead of the correct one.
   e.g. typed 'r' instead of 'e' in 'hello' and produced 'hrllo'.
4. EXTRA ON LEFT (l): The volunteer pressed the correct key and an "adjacent" key together and the wrong letter was typed first.
   e.g. pressed keys for 'l' and 'i' when typing 'l' in 'hello', producing 'heillo'.
5. EXTRA ON RIGHT (r): The volunteer pressed the correct key and an "adjacent" key together and the correct letter was typed first.
   e.g. pressed keys for 'h' and 'y' when typing 'h' in 'hello', producing 'hyello'.

A key is "adjacent" to another if they share a border or a corner. See the image below.



Adjacent keys of 'z', 'g' and 'o'

As an example, if the reference word is 'normal' and the output word is 'nrtmsap', there is only one sequence of the above typos that could produce that output word ofrolw, because:

- 'n' is ok (o)
- 'o' is missing (f)

- 'r' followed by its adjacent 't' (r)
- 'm' is ok (o)
- 'a' preceded by its adjacent 's' (l)
- 'l' replaced by its adjacent 'p' (w)

There may be more than one sequence of typos that lead to the same output word.

As an example of such scenario if while typing 'allok', the volunteer produced 'allok' (yes, the same word, sequence ooooo), there can also be a sequence of typos that happen to produce the same word. ofowl, for example:

- 'a' is ok (o)
- 'l' is missing (f)
- 'l' is ok (o)
- 'o' replaced by its adjacent 'l' (w)
- 'k' preceded by its adjacent 'o' (l)

Input

The first line will only have a single positive integer K, smaller than 50. Then follow K pairs of words (each word in its own line), the first being the reference word and the second being the output word. Look at the sample.

The words are only formed by at most 15 lower case letters and there's always a sequence of typos that, when followed while attempting to type the reference word, can produce the output word.

While the reference word must have at least 1 character, the output word may be empty (since you can always forget to type all the letters).

Output

The output of each test case must be followed by a blank line. The output of a test case starts with a header on a single line, which consists of the reference word, followed by the output word, separated only by a single space, and finished with a colon (:). Look at the sample.

After the header, list on independent lines each possible sequence of typos that the volunteer could have made to type the output word while attempting to type the reference word.

Sequences must be ordered lexicographically.

Example

Input	Output
3 allok allok empty normal nrtmsap	allok allok: ofolo ofowl ooflo oofwl ooofl ooooo ooorf oorfo oorwf empty : fffff normal nrtmsap: ofrolw