Problem C. Correcting Curiosity

Input file:	curiosity.in
Output file:	curiosity.out
Time limit:	2 seconds
Memory limit:	256 megabytes

Curiosity is the rover that explores the Gale Crater on Mars. Recently it found an evidence of water in Martian soil, which will make it easier to plan the future manned missions.

Curiosity can communicate with Earth directly at speeds up to 32 Kbit/s, but on average 14 minutes and 6 seconds will be required for signals to travel between Earth and Mars.

"You have just seen a stone and applied brakes, but you know that the rover is already passing that stone" — Matt Heverly, the rover's driver, explains. "So we just plan the route, then write down a list of simple textual commands: move one meter ahead, turn left, make a photo and so on".

Sometimes it is necessary to react very fast to some unexpected events. For example, if the cameras have seen something interesting, then you might want to change the route of the rover to make an additional photo. To do that, you send a substitution command of the form `s/`⟨*string*⟩`/`⟨*replacement*⟩`/g`. This replaces all occurrences of ⟨*string*⟩, starting with the leftmost one, to ⟨*replacement*⟩.

More formally, if A is a non-empty string and B is a string, then to apply the substitution command `s/`A`/`B`/g` to a string S, you should do the following:

1. Find the leftmost occurrence of A in S, such that $S = S_L + A + S_R$.
2. If there is no such occurrence, stop. Then, S is the answer.
3. Let R be the result of applying `s/`A`/`B`/g` to S_R.
4. The answer is $S_L + B + R$.

This means that:

1. If there are two overlapping occurrences of A in S, only the leftmost one is replaced. For example, applying "`s/aba/c/g`" to "`abababa`" yields "`cbc`": after replacing the first occurrence of "`aba`" the string turns to "`cbaba`", and only the last occurrence of "`aba`" can be replaced after that.
2. No substitution uses the results of previous substitutions. For example, applying "`s/a/ab/g`" to "`a`" yields "`ab`", applying "`s/a/ba/g`" to "`a`" yields "`ba`".

You know that the longer is the command, the bigger is the time necessary to transmit it. So, you have to write a program that finds shortest command that transforms the initial string to the final string.

Input

The first line contains the initial and the final strings. Both strings are non-empty and their lengths do not exceed 2000 characters. The strings contain only English letters, spaces and punctuation signs (commas, colons, semicolons and hyphens: '`,`', '`:`', '`;`', '`-`'). The given strings are not equal.

Output

Output the substitution command that transforms initial string to final string and has the minimum length. If there are several shortest substitution commands, output any of them.

Examples

curiosity.in	curiosity.out
move left, move right; move up move left, move down, move up	s/right;/down,/g
If not found: move x; else move -x If found: move x; else move -x	s/ not//g
abababa cbc	s/aba/c/g