F Fraud Checking

Time limit: 1s

Image by Maarten Sijm. Magnifying glass from Wikimedia Commons.

The exam of the First Programming Course (FPC) was a success: all students have passed, with excellent grades! In fact, they have done so well, that it is somewhat suspicious. It is time to do a fraud check, but nobody has time to sift through the hundreds of code submissions manually. Therefore, you have been tasked with finding code submissions that are similar in structure: you should make a fraud report if the only difference between two pieces of code is the renaming of some variables.

More precisely, one piece of code is *similar* to another when it is possible to make the two pieces of code identical to each other by processing a sequence of word replacements. When, after processing all word replacements, the first piece of code is identical to the second piece of code, and this also applies vice versa, then the two pieces of code are similar.

Each word replacement should replace *all* occurrences of a word. A word replacement can only replace a *complete* word. Spaces between words or at the start/end of a line should be ignored, but newlines cannot be ignored: for two pieces of code to be similar, each line of the first piece should have the same number of words as the same line in the second piece.

Given two pieces of code, check whether they *similar*. If they are similar, print a fraud report that gives the sorted list of word replacements.

Input

The input consists of:

- A line with one integer n ($1 \leq n \leq 1000$), the number of lines of each code submission.
- n lines of code for the first submission.
- n lines of code for the second submission.

The lines of code consist of only spaces (‘ ’), letters from the English alphabet (‘`A-Z`’ and ‘`a-z`’), numbers (‘`0-9`’), and underscores (‘`_`’). The lines of code are at least 1 and at most 120 characters long, excluding the newline character.

Output

If the two pieces of code are not similar, output “`-1`”.

If the two pieces of code are similar, output:

- A line containing a non-negative number r, indicating the number of word replacements.
- r lines, each containing a word replacement. A word replacement consists of two words, separated by a space. The list should be lexicographically ordered by ASCII values.

Sample Input 1

```
4
i is 42
while i
    subtract i
print i
j is 1337
while j
    add j
print j
```

Sample Output 1

```
3
42 1337
i j
subtract add
```

Sample Input 2

```
3
ans is 42
if ans is 42
    print ans
ans is 42
if ans is not 42
    print ans
```

Sample Output 2

```
-1
```

Sample Input 3

```
2
ans is 42
print items
ans is 42
print ans
```

Sample Output 3

```
-1
```