Ever so often on Codefights, a user tries to submit a duplicate solution they copied from someone else. Generally these are pretty easy to detect and block. However, it gets trickier when you have a duplicate solution with some variables renamed to avoid getting caught.

The cheating usually happens as follows: in a text editor the *"Find and replace"* function is applied to all occurrences of some variable name `A` that consists of letters, digits, underscores, and starts with a non-digit character (since it's a variable name), to change it to some other variable name `B` that fulfills the same constraints.

It would appear that after applying this *"Find and replace"* procedure **multiple times** it would be impossible to detect duplicates, but this isn't the case. Your goal is to implement an algorithm that compares two code snippets and determines whether one of them could be produced from the other using the above-described approach.

Note. Here is a formal definition of how *"Find and replace"* function works. When searching for string `A` to replace all of its occurrences in string `S` with string `B`, it first finds the leftmost occurrence of `A` in `S`. Then it replaces this occurrence of `A` with `B`. Then it repeats the above procedure for the suffix of `S` which starts immediately after the last character of the inserted copy of `B`. The process repeats until the remaining suffix contains no occurrences of `A`.

Example

- For

```
code1 = ["def is_even_sum(a, b):",
         "    return (a + b) % 2 == 0"]
```

and

```
code2 = ["def is_even_sum(summand_1, summand_2):",
         "    return (summand_1 + summand_2) % 2 == 0"]
```

the output should be `plagiarismCheck(code1, code2) = true`.

All occurrences of `a` are replaced with `summand_1`, and all occurrences of `b` are replaced with `summand_2`.

–