CODE PLAGIARISM

CHALLENGE DESCRIPTION:

We propose you to solve an exciting challenge: find plagiarism in one piece of source code against another. This challenge is similar to revealing plagiarism in two texts; but, in case of source code, we know the syntax of a language and we can be more accurate in finding renamed data structures and other cheating.

We do not provide any ranking points for this task. However, those developers who send a good working solution to us will receive a special badge.

To make the challenge more interesting, we have chosen two programming languages: Python and Go. You will have two pairs of source code to compare. The first one is written in Python, and the second one – in Go. You will need to find duplicates in Python code chunks, and then in Go code chunks. Thus, you will not need to compare Python code and Go code.

Our test cases for both languages are the following:

1. The same code
2. Changed names of variables
3. One is a half of another
4. Removed documentation strings and comments
5. Shuffled functions
6. Small part of code is inside of another code
7. Two approaches to the same problem
8. Different code
9. "Hello World"

Please feel free to contact the support at codeeval.com if you can add more test cases suitable for this task. Also, if for successful challenge completion you need to install some library for the language used, contact the support.

INPUT SAMPLE:

Your program should accept a filename as its first argument. The input file contains 18 sets of test cases (9 for Python and 9 for Go). You can check the input file on GitHub. Each test case is separated by five asterisks. The first line of each test case is the title, for example "<<< Python The same code." Two code samples inside of each test case are separated by five equal signs.

For example:

```
<<< Python The same code   # first test case

def quickSort(alist):
    ... new code

=====

def quickSort(alist):
    ... existing code

*****
<<< Go The same code  # second test case

package main
    ... new code

=====

package main
    ... existing code

*****
<<< Python Changed names of variables  # Third test case
            etc.
```

OUTPUT SAMPLE:

For each test case, produce a single line of output containing the percentage of plagiarism comparing *new* code with the *existing* one.

```
100
95
0
```