- **About**

- Ved Bhatawdekar:
- **Home**
- **Logout**

Coding Contest Round 2

You are at question 1 of 1.

Almost Duplicates

Consider an input file that contains customer records of a company - *i.e.* names and addresses of US-based customers - in the following format:

```
David Naughton
1210, West Dayton Street, #7
Madison, WI-53717
--
Rajesh Malhotra
133, Circle Ct.
Marietta, GA-13325
--
Rajesh Malhotra
133, Circle Ct.
Marietta, GA-13325
--
James Butt
6649 N Blue Gum St
New Orleans LA-70116
--
Josephine Darakjy
4 B Blue Ridge Blvd
Brighton, MI-48116
--
Josephine Darakjy
5 B Blue Ridge Blvd
Brighton, MI-48116
--
Josephine Darakji
4 B Blue Ridge Blvd
Brighton, MI-48116
--
James But
6649 N Blue Gum St
New Orleans LA-70116
--
David Naughton
1210 West Dayton Street
Madison, WI-53717
--
Art Venere
8 W Cerritos Ave #54
Bridgeport, NJ 80141
```

Different records in this file are separated by a line containing just `--`. Each record must be exactly 3 lines. The first line or a record must be a name, the second line must be a street address, and the third line is a city name followed by a ZIP code.

Unfortunately, as is usually the case with such databases, there are a number of duplicates, or "almost duplicates". You have to write a program that will find which records are duplicates or almost duplicates of other records in this file.

Specifically, two records are considered almost duplicates of each other if the following conditions hold:

- For the purposes of this matching, all whitespace is first standardized as follows: leading and trailing whitespace on any line is ignored, and within a line, multiple consecutive whitespaces are considered as one space. Thus, "`John Paul`" matches "` John Paul`" and "`John    Paul`" but not "`JohnPaul`".
- After whitespace standardization, the two records should have a maximum of two mismatches, a maximum of one per line. A mismatch is defined as:
  - A single character less (*i.e.* deleted from one of the records)
  - A single extra character (*i.e.* inserted into one of the records)
  - A single character different
- If two records have more than 2 mismatches, or if they have 2 mismatches on the same line, they are not considered almost duplicates.

You have to read input from a file called `input.txt` (in the current directory) count the number of records in the input file that are duplicates or almost duplicates and write the total number to a file called `output.txt` (in the current directory)

Thus, if the `input.txt` contained the data give in the example above, your `output.txt` should contain just:

```
7
```

Because there are 2 "`Rajesh Malhotra`", 2 "`James Butt`"s, and 3 "`Josephine Darakjy`"s. Note: the two "`David Naughton`"s are not almost duplicates because there are more than 1 mismatches in the 2nd line.

Note the following simplifications for the purposes of this contest:

- Do not try to match short-forms of names or parts of addresses. Thus, "`David`" does not match "`Dave`" and "`Park St.`" does not match "`Park Street`".
- Also, you're not expected to parse/understand the actual content of the names or addresses.
  - *e.g.* As far as you're concerned, "`Apt #001`" is not the same as "`Apt #1`"
  - *e.g.* It is not an error for the name line to contain characters not normally found in a name, or for the address lines to not match what a normal address lines look like.

Any records that don't consist of exactly 3 lines should be ignored.

Input and Output files

- Your program will not be accepted by the system unless it reads from the right input file and writes to the correct output file
- Your program must read the input from a text file called `input.txt` in the current directory. Not `C:\input.txt` or any other path like that. If you can't figure out how to read input from `input.txt` in the current directory, you will not be able to solve this problem.
- Your program *must* write output to a file called `output.txt` in the current directory. (Current directory is the directory in which your program is executed at runtime.) The `output.txt` must contain just the expected output (in the exact format as given above) and nothing else.

Important Notes

- Your program must be able to work with any other inputs that are in the same format.
- No hardcoding. *Do not simply write a program that outputs the expected answer to the output file without actually doing the calculations. If you do that we will detect it and your entire entry will be disqualified.*
- If you write your program in `C/C++`, please use *standard* `C/C++`. Do not use platform specific features like `conio.h` or `clrscr` or `getch`. **Click here for detailed instructions**.
- Java programmers, the system expects that your program will contain a class called `Main` and that class will have the `main` method. Also do not put a `package` declaration in your program. **Click here for detailed instructions**.

Remember: this problem seems simple until you start worrying about all the corner cases, and the efficiency of your program, and being able to handle extreme conditions.

Answer

Language/Platform

python3

Submit

- *Note: this is a blocking question. If you're not able to provide a correct answer, you can not proceed further. You can log off at any time. Your work so far has been recorded in the system, and you will be evaluated on the basis of that.*

Powered by:

Credits: Icons taken from **Silk icons** and **iconspedia.com**.