We at ecobee care about the environment. So much so that being planet positive is one of our core values. We also like to help our customers to understand their impact on the environment and how they can help to improve and lower their energy consumption. One very important parameter is how well their home is insulated. This parameter is called R-value. A well-insulated home requires less runtime of the heating/cooling equipment and has a high R-value; a not so well insulated home has a low R-value.

In this task, you are to write a program that will rate the ecobee thermostat owners by the quality of insulation of their home. Each user's R-value is to be rated relative to the others in the same region (city/province/county) they live.

The rating is a number between 1 and 10. For a given home and region, if the percentage of homes with better insulation (R-value) is in the [90, 100) range, then the home is rated 1, if the percentage of homes with better insulation is in the [80, 90) range, then the home is rated 2. A home that is in the top range, such that percentage of homes with better insulation falls in [0, 10) range, it is rated 10.

The Input data is provided on stdin and the program should write to stdout.

The **Input** consists of two sections, a *data section* followed by an empty line, followed by a *query section*.

The *data section* is a sequence of lines in the following format:

"<name>" "<location>" <rvalue>

"<name>" "<location>" <rvalue>

...

where:­­­­­­­­­­

* <name> is the customer's name
* <location> is the location of the building and has the following format: <country>/<state>/<city>
* <rvalue> is R-value for the building, a number between 0.0 and 50.0, with at most three digits after the decimal point.

The lines in the *data section* are not in any particular order.

The *query section* is a sequence of lines containing pairs of name and region, one pair on each line. You are to calculate the rating for each pair in the query section. The format is:

"<name>" "<region>"

"<name>" "<region>"

...

where:

* name is customer's name
* region can be one of <country>, <country>/<province> or <country/province/city>

There will be no more then 1,000,000 locations in the *data section* (across all regions). You can assume that any pair that appears in the *query section* is valid and rating can be calculated for it. There will be no duplicate pairs in the query section. The lines are not in any particular order.

The **output** should contain the same number of lines as the *query section* of the input, in the following format:

"<name>" "<region>" "<score>"

where:

* name is customer name
* region is a region in which the customer is scored
* score a number from 1 to 10 that represents the rating of the building in that region.

The output should be in the same order as appears in the *query section* of the input.

Sample input:

"John Doe" "Canada/Ontario/Toronto" 1.5

"Samanta Smith" "Canada/Ontario/London" 3.7

"Adam Xin" "Canada/British Columbia/Vancouver" 2.110

"Monica Taylor" "Canada/Ontario/Toronto" 2.110

"Alicia Yazzie" "US/Arizona/Phoenix" 5.532

"Mohammed Zadeh" "Canada/Ontario/Toronto" 1.43

"John Doe" "Canada"

"John Doe" "Canada/Ontario"

"Alicia Yazzie" "US/Arizona"

Sample output:

"John Doe" "Canada" 4

"John Doe" "Canada/Ontario" 5

"Alicia Yazzie" "US/Arizona" 10

Other considerations:

* You can write your program in any general-purpose language, though we have a bias towards Java as we use that ourselves
* You can use unit-test library/framework for your unit tests

What we value:

* Correctness - We will run the program against inputs and match with predefined outputs. The program should parse the input as defined above and should produce correct output.
* Good design - you should structure the program in units/modules with well defined purpose, but be wary of over-engineering / designing. YAGNI and DRY are things we believe in.
* Unit testing - We expect to have meaningful unit tests that cover various cases, with good code coverage, but also focused on edge cases
* Performance - your program should be efficient, both in time and in space
* Please include instructions for building your project on the command line - this is necessary for your submission to be evaluated