Counting forest data X79058_en

We have information about things we have found in the forest:

- Name (`string`)
- Color (`string`)
- Is it poisonous? (`string`)
- Does it burn? (`string`)
- Height (`double`)

Write a program that reads a vector of these things, and allows the user to count how many things satisfy several *queries*. A query consists of five elements: four strings (for querying the name and the color, whether it poisons or it burns), and a double for defining the threshold height. Depending on the field type, queries can be incomplete: `string` can have a value or a wildcard `"*"` to denote any value. For instance, the following query

```
* red F * 12.4
```

asks for counting how many red non-poisonous things exist in the forest that are higher or equal to 12.4 centimetres.
You are required to extend the following program, WITHOUT modifying or adding anything outside of the indicated places.

```
#include <iostream>
#include <vector>

using namespace std;

struct Things {
    string name;
    string color;
    string poison;
    string burns;
    double height;
};

typedef vector<Things> Forest;

void read_data(Forest& v) {
    // ADD YOUR CODE HERE
}

int count(const Forest& v, string name, string color,
          string poison, string burns, double height) {
    // ADD YOUR CODE HERE
}
```

```
int main() {
    Forest v;
    read_data(v);

    // ADD YOUR CODE HERE
}
```

Exam score: 3.000000 **Automatic part:** 100.000000%

Input

A natural number $n \geq 0$, followed by the information of n things observed in the forest. Each thing is represented as four strings and a double.
After that, a set of queries is introduced, where each query is represented as four strings and a double, as shown in the examples.

Output

For each query, the number of things that match the query.

Sample input

```
4
poma vermell F F 6.5
fulla marro F T 2.2
mora vermell T F 0.5
branca marro F T 34.3

poma vermell F F 6.5
*  vermell * * 0
*  vermell * * 4
*  *  F * 2.1
*  * T  * 0
```

Sample output

```
1
2
1
3
1
```

Problem information

Author : Professors de Pro1
Generation : 2018-12-21 11:49:45