Problem A. Random Access Memory

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	256 megabytes

The notice on the screen displayed for a few hours already: "User Alice has left you a message. Play it?" Chris was in no hurry: he needed to be in special mood to respond to a voice message. Finally, he leaned back in his chair, pressed a few keys with a usual movement and closed his eyes.

— Hello, my name is Alice. I represent the "Stars of information" company, we were given very good recommendations about you and we have an interesting offer for you! — high female voice chirped, — Your students may be interested in our offer. We are a young company...

"Your students" — that was unexpected. Chris taught several courses at the Faculty of Mathematics and Natural Sciences in the University, and he was sure that he knew all companies, who could be interested in the graduates. He noticed that the name of the company was quite strange, but listened more carefully.

Alice told that the company needed young and ambitious software engineers, who had good education, that the company was ready to give wonderful conditions for students. And finally, that she, Alice, invites Chris to the office — to look at the well-equipped workplaces and recreation rooms. For instance, he could come tomorrow at half past three in the afternoon. But Alice didn't say what was software company developing and what tools and IDE were used.

However, Chris was pretty sure that he knew the chief-executive of the company. He looked at the map: apparently, the office was located near Chris's house. Also, tomorrow was Tuesday, and he was free after three on Tuesday. He thought once again that it was someone he surely knew, so Chris said "OK. I'll be there" and sent this message to Alice.

On the way home he was thinking, who could be the chief-executive of this company, what kind of software this company was developing, and what programming language was primary during development.

Chris bases his assumptions as triples: (s_i, p_i, l_i), where s_i is the name of one of his acquaintances, p_i — kind of software that the company could be developing, and l_i — the programming language in use. However, for every assumption he soon finds refutation, because he remembers that one of the three elements should be excluded from consideration. Please note that, for example, Haskell as a programming language and Haskell as a name of one of Chris's acquaintances are different (i.e. they are two different elements).

While Chris thinks of the next triple, he forgets all previous triples and refutations completely. Therefore, even if he already found refutation for some element, he can again include this element in the next triples.

Chris already considered N triples and found a refutation for each one. You are given a sequence of these triples and their refutations. After each triple you have to report how many elements are not refuted at each of the three positions at the moment. The element at a position is considered to be not refuted if Chris assumed this element in this position before, but have not yet found a refutation for this element in this position.

Input

The first line contains an integer N ($1 \le N \le 100$) — the number of the triples with their refutations.

Each of the following N lines contains a triple with its refutation. Each element of the triple is a non-empty string of up to 50 characters containing lowercase letters. All three elements are distinct and separated by a space. Then, after a space, there is a refutation which coincides with one of the three elements.

Output

Output contains N lines.

In each line, output three integers, separated by a space: how many elements are not refuted at the first, second and third position respectively.

Examples

input.txt	output.txt
3 gates windows pascal gates gates windows pascal windows gates windows pascal pascal	0 1 1 0 0 1 0 0 0
2 haskell matrix pascal haskell pascal matrix haskell haskell	0 1 1 1 1 1
3 carmack fallout cpp carmack carmack fallout cpp carmack carmack fallout cpp carmack	0 1 1 0 1 1 0 1 1