Frank lives in London and he really likes games and maths. Lately, he has been playing a game on his mobile phone. It is simple, a sequence of coloured tokens is provided and each turn the player has to combine a pair of adjacent tokens to obtain a new token of certain colour. This process is repeated until the whole sequence results in only one final token. The problem is that not every pair of colours can be merged. There is a set of rules describing the allowed combinations. For example, given the following rules

blue	+	yellow	→	green
yellow	+	red	→	orange
blue	+	orange	→	brown

and the sequence (`blue`, `yellow`, `red`), the game could be finished obtaining a `brown` token after two steps: (`blue`, `yellow`, `red`) → (`blue`, `orange`) → (`brown`).

Frank is now in Valencia attending a famous programming contest and he is waiting for the tram to go to the university. Meanwhile, he is playing this game to fill time but the sun is shining so bright that he can not properly see the screen of his phone. He has some certainty about the possible colour of each token and he is wondering what will be the resulting colour following the most likely play of the game according to his estimation of the input sequence. Given Frank's estimation of two colours `A` and `B`, and the combination `A` + `B` → `C`, the certainty of the obtained colour `C` is computed as $\mathsf{cer}(\mathtt{C}) = \mathsf{cer}(\mathtt{A}) \cdot \mathsf{cer}(\mathtt{B})$.

INPUT

The first line contains R ($0 < R \leq 100$) the number of rules describing the allowed combinations of colours. Each of the following R lines contains three strings s_1, s_2, s_3 representing the combination $s_1 + s_2 \rightarrow s_3$. Next line shows the number of test cases T. For each test case an integer C indicates the length of the input sequence of tokens ($0 < C \leq 500$). Each of the next C lines describes a token, it contains a list of pairs (k, $\mathsf{cer}(\mathsf{k})$) providing a colour k and its corresponding certainty ($0 < \mathsf{cer}(\mathsf{k}) \leq 1.0$). The list ends with the word `END`. The sum of the certainties for a certain token always is 1.0. Every colour in a test case has appeared first in the rules describing the game.

OUTPUT

For each test case, print the resulting colour of the most likely play of the game. If there is no possible combination to finish the game then print `GAMEOVER`.

SAMPLE INPUT

```
7
Blue Yellow Green
Yellow Red Orange
Green Red Yellow
White Red Pink
Pink Black Red
Orange Red Red
Yellow Orange Yellow
3
2
Red 0.7 Orange 0.3 END
Yellow 1.0 END
4
Blue 0.6 Green 0.4 END
Red 0.2 Orange 0.6 Yellow 0.2 END
White 0.9 Yellow 0.1 END
Red 0.5 Black 0.5 END
2
Blue 1.0 END
Orange 1.0 END
```

SAMPLE OUTPUT

```
Orange
Yellow
GAMEOVER
```

EXPLANATION OF THE SAMPLES

First case

In the first case there are only two tokens. Frank is sure that the second token is `Yellow`, but the first token could be either `Red` or `Orange`. There are two possible solutions to the game:

1. (`Red`, `Yellow`) → (`Orange`) :

   cer(`Orange`) = 0.7

2. (`Orange`, `Yellow`) → (`Yellow`) :

   cer(`Yellow`) = 0.3

Hence, according to Frank's estimation the most likely play is the number 1 such that the final colour is `Orange`.

Second case

In the second case there are several tokens and estimations. Two possible solutions are:

1. `(Blue,Yellow,Yellow,Red)` → `(Blue,Yellow,Orange)` → `(Blue,Yellow)` → `(Green)`

   cer(`Green`) = 0.006

2. `(Green,Red,White,Black)` → `(Green,Pink,Black)` → `(Green,Red)` → `(Yellow)`

   cer(`Yellow`) = 0.036

Hence, according to Frank's estimation the most likely play is the number 2 such that the final colour is `Yellow`.

Third case

In this final case Frank is sure that there are two tokens `Blue` and `Orange`. There is no rule in the defined game that allows the combination of those colours, hence, there is no solution for this sequence of tokens.

(This page is intentionally left blank)