Problem B. Bigram Language Model

Input file:	`stdin`
Output file:	`stdout`
Time limit:	5 seconds
Memory limit:	256 MB

In natural language processing, language models gives how likely a certain English sentence appear in a context (e.g. casual conversations, academic papers, or news websites) by assigning a probability distribution over all possible sentences. Bigram language model approaches this modeling problem by estimating the transition probabilities from previous word to next word. Formally,

$$P(S = w_1 w_2 \dots w_m) = P(w_1)P(w_2 \mid w_1) \dots P(w_m \mid w_{m-1})$$

We can estimate transition probabilities from word s to t from a corpus (a collection of sentences) collected from the context:

$$P(t \mid s) = \frac{c(s,t)}{\sum_{t' \in V} c(s,t')}$$

where V denotes the vocabulary (aka set of all words in the corpus), and $c(s,t)$ denotes the total number of times that word t comes right after word s in the same sentence in the corpus.

Suzukaze has collected a corpus and is trying to compute the probability of some sentences. He needs your help to get some transition probabilities. Can you help him?

Input

The first line contains an integer n ($1 \le n \le 1000$), the number of sentences in the corpus.

In the following n lines, the i-th line starts with an integer m_i ($1 \le m_i \le 100$), the number of words in the i-th sentence. It is then followed by m_i space-separated words.

The next line contains an integer q ($1 \le q \le 10^4$), the number of queries.

In the following q lines, each line contains two space-separated words s and t, querying for the estimated transition probability from word s to t.

All words in the corpus and queries are no more than 10 characters long and contain lowercase letters only.

Output

For each query, output a line containing the estimated transition probability for the queried word pair. Print the number as an irreducible fraction. (See example for details)

If you cannot estimate the transition probability from the corpus, print "Insufficient data"instead.

Examples

stdin	stdout
5 7 get busy living or get busy dying 4 stay hungry stay foolish 6 whatever you do do it well 6 everything you can imagine is real 5 the things you can find 8 get busy busy living hungry stay foolish stay do do you do you can can do	1/1 1/2 1/1 Insufficient data 1/2 1/3 2/3 0/1