Problem A. Bigram Language Model

Input file: stdin
Output file: stdout
Time limit: 5 second

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:

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

where $c(s,t)$ 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 a real number - the estimated transition probability for the queried word pair. Always print 4 digits after the decimal point.

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 find	1.0000 0.5000 1.0000 Insufficient data 0.5000 0.3333 0.6667 0.5000

Problem B. Fruit on the Tree

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 second

“Triangoes”, a new type of fruit, are in triangular shapes and taste like mangoes. However, nobody in the world has ever seen “triangoes” since they always grow implicitly on the tree and hide in the triangles. Formally, a “triango” tree is an undirected weighted tree and a “triango” is a set of three vertices on the “triango” tree such that the lengths of the three paths between each pair of these three vertices satisfy the triangle inequality; that is to say, they form a triangle.

After a long expedition, **Suzukaze** has eventually found a “triango” tree in his house. He needs your help to count the number of “triangoes” on the “triango” tree. Can you help him?

Input

The first line contains an integer n ($1 \le n \le 10^5$), the number of nodes on the “triango” tree. The nodes on the “triango” tree are labeled from 1 to n.

In the following $n-1$ lines, each of the lines contains 3 integers u, v, w ($1 \le u, v \le n; u \ne v; 1 \le w \le 10^5$), which means that there is a weighted edge whose weight is w connecting node u and node v.

It’s guaranteed that the input data is a tree.

Output

Output an integer - the number of “triangoes” on the “triango” tree.

Examples

stdin	stdout
7 1 2 1 1 3 1 2 4 1 2 5 1 3 6 1 3 7 1	8

Problem C. Greenberg Mass Comparison

Input file: stdin
Output file: stdout
Time limit: 1 second

Linguist Joseph Greenberg proposed the method of mass comparison for determining genetic relatedness between languages. In this method, N languages are categorized into one or more families. Formally, given a set of N different languages $\mathcal{L} = \{L_1, \ldots, L_N\}$, a relation analysis is a set of families $\mathcal{F}$, satisfying the following properties:

- $\mathcal{F} = \{F_1, \ldots, F_k\}$ for some k
- For $1 \le i \le k$, $F_i = \{L_{i,1}, \ldots, L_{i,m_i}\}$ for some m_i
- For $1 \le i, j \le k, i \ne j$, $F_i \cap F_j = \varnothing$
- $\cup_{1 \le i \le k} F_i = \mathcal{L}$

Greenberg wants to know how many distinct relation analysis are there for N languages. Two relation analyses $\mathcal{F}_1$ and $\mathcal{F}_2$ are distinct if $\mathcal{F}_1 \ne \mathcal{F}_2$. Can you help him compute this number?

Input

The first line of input contains a single integer T ($1 \le T \le 100$), the number of test cases. In the following T lines, each line contains a single integer N ($1 \le N \le 100$).

Output

For each test case, output a line containing the answer for the queried N.

Examples

stdin	stdout
4 1 2 3 40	1 2 5 157450588391204931289324344702531067