Task: TAK
Taxis

Olimpiada Informatyczna

XXV OI, Stage III, Day two. Source file `tak.*` Available memory: 256 MB. *12.04.2018*

Byteasar is the editor-in-chief of the *Your taxi* magazine, which publishes annual rankings of the best taxi companies in Byteotia. The preparations for the most recent edition, focusing on the most affordable taxis, are about to begin.

The cost of a ride provided by the i-th taxi company is always determined as the sum of:

- a fixed charge s_i for taking a taxi, independent of the travel distance,
- a variable distance charge: the travel distance of d bytemeters is multiplied by the charge per bytemeter c_i.

Each company has its own parameters s_i and c_i.

Byteasar has a strong opinion on the true ranking of taxi companies, and in past rankings he has taken multiple criteria into account, carefully chosen so that the resulting "objective" ranking matched his beliefs. Harsh criticism forced him to simplify the rules: this year's edition will focus on the ride price, which most readers find objective. However, Byteasar can still tweak one parameter, so that the final ranking matches his conviction: the travel distance d, which has to be *positive* (though not necessarily integer). Naturally, Byteasar is allowed (by his own conscience) to break ties in an arbitrary way.

That said, Byteasar's opinion is firm yet ever changing, and for good reasons: Not only do the taxi companies offer Byteasar bribes, they also constantly change the quality of their services, necessitating changes in the ranking. Write a program that will find the appropriate distance travel d after each change of conditions.

Input

In the first line of the standard input, there is a positive integer n which specifies the number of taxi companies. In the n lines that follow, successive companies are described by their parameters: each line contains two non-negative integers s_i and c_i, separated by a single space, which are: the fixed charge and the variable distance charge per bytemeter of the i-th company.

The next line contains a sequence of n pairwise different positive integers ranging from 1 to n, which is Byteasar's initial ranking (the i-th element in the sequence is the number of the company ranked i-th).

The next line contains a single non-negative integer q which specifies the number of updates. Then q lines follow, describing successive updates: each line contains two distinct positive integers a_i and b_i, which indicate that Byteasar would like to swap the companies currently ranked a_i-th and b_i-th.

Output

Your program should print exactly $q + 1$ lines to the standard output. The i-th line should contain a single positive rational number: the travel distance which makes the ranking by the price rule match the one desired by Byteasar after $i - 1$ updates. The distance should be printed as an irreducible fraction x/y such that neither the numerator x nor the denominator y exceed 10^9.

If there is no such distance, the word `NIE` (Polish for *no*) should be printed instead.

Example

For the following input data:

```
3
8 3
12 2
9 4
2 1 3
3
1 3
1 2
2 3
```

a correct answer is:

```
4/1
NIE
1/1
2/1
```

Explanation for the example: To get 2, 1, 3 as a ranking of companies, Byteasar should set the distance to $d = 4$. Then the total travel costs are respectively $8 + 3d = 20$, $12 + 2d = 20$, and $9 + 4d = 25$. Since companies no. 1 and 2 charge the same, Byteasar can choose their order in the ranking. After swapping companies ranked 1st and 3rd, we get a ranking 3, 1, 2, which cannot be attained, regardless of the value of d. After the next update, the ranking is 1, 3, 2; using $d = 1$ yields conforming costs 11, 14, and 13. After the final swap, the ranking is 1, 2, 3, which can be attained for $d = 2$: then the costs are 14, 16, and 17.

Sample grading tests:

`1ocen:` $n = q = 10$, a random test conforming with subset 4;

`2ocen:` $n = q = 10$, a random test;

`3ocen:` $n = q = 10$, the cost of traveling distance $d = 1/3$ is the same with all taxi companies;

`4ocen:` $n = q = 1000$, only one ranking is possible: for every value of the parameter d, the 1st company is cheaper than the 2nd, which is cheaper than the 3rd, and so on. This is the ranking initially desired by Byteasar. Afterwards, in every odd update Byteasar swaps two positions in this ranking, only to swap them back again in the following even update, recovering the initial ranking.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several unit tests.

The following hold in all tests: $1 \leq n \leq 500\,000$, $0 \leq q \leq 500\,000$ i $1 \leq s_i, c_i \leq 10^9$.

Subset	Property	Score
1	$q = 0$, no simultaneous ties	10
2	$n, q \leq 2000$, no simultaneous ties	10
3	$n \leq 2000$, no simultaneous ties	25
4	no `NIE`	30
5	no additional constraints	25

In above table, "no simultaneous ties" means that for all positive values of d, there is at most one (distinct) pair of companies whose total charge for travel distance d is the same. Moreover, "no `NIE`" means that the correct output does not contain the word `NIE`.