Task: TUR
Triinformathlon

XXV OI, Stage III, Day one. Source file `tur.*` Available memory: 128 MB. *11.04.2018*

Byteotian informaticians are well respected in their country. So much so, that a TV show titled *Triinformathlon* has been enormously popular for years. In this year's edition, there are n contestants (assigned starting numbers from 1 to n) competing in three informatics disciplines (this year: timed implementation of suffix trees, debugging the SIO2 system, and taking the Turing test). For each discipline, full standings have been announced, so the rank of each contestant is known in each discipline. Sophisticated tie-breaking rules ensured there were no ties in any discipline.

Every Byteotian is rooting for some contestant, and discussions – often heated, especially on social media – whether one contender is better than another, are a favorite pastime. The fact that "being better" is not clear-cut with three independent disciplines makes such discussions all the more engaging.

Byteasar spotted a golden opportunity: an app which compares the contestants should prove extremely popular. To make the comparisons possible, he introduced the following relation:

*A contender a **morally dominates** contender b if at least one of the following conditions holds:*

- *in at least two out of three disciplines, a is ranked better than b, or*
- *there exists a contender c such that a morally dominates c, and c morally dominates b.*[†]

Byteasar's start-up is overwhelmed with other projects, so implementing a comparison algorithm has been commissioned to you. Write a program that, based on the standings for all three disciplines, will answer m queries of the form "Does contender a morally dominate contender b?".

Input

In the first line of the standard input, there is a single integer n ($n \geq 2$) which specifies the number of contenders.

In the second line, there is a sequence of pairwise different integers from the range $[1, n]$, separated by single spaces. These are the ranks of successive contestants in the first discipline. The third and fourth line contain similar sequences, specifying the ranks in the second and third discipline in the same format.

The fifth line contains a single integer m ($m \geq 1$) which specifies the number of queries. The following m lines give those queries: the i-th line contains two integers a_i and b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), separated by a single space, which stand for a query "Does contender a_i morally dominate contender b_i?".

Output

Exactly m lines should be printed to the standard output: the i-th of them should contain a single word `TAK` (Polish for *yes*) if the answer to the i-th query is positive or `NIE` (Polish for *no*) in the opposite case.

Example

For the input data:

```
5
1 2 4 3 5
2 3 5 1 4
3 1 5 2 4
4
2 4
4 2
1 5
5 1
```

the correct result is:

```
TAK
TAK
TAK
NIE
```

[†] Formally, we define a set M as the *minimum* (inclusion-wise) set that contains all pairs (a, b) such that a is ranked better than b in at least two disciplines, and such that if (a, c) and (c, b) are in M, then so is (a, b). Then we say that a morally dominates b if (a, b) is in M.

Explanation for the example: Contender no. 2 morally dominates contender no. 4, because he is ranked higher in the first and the third discipline. On the other hand, contender no. 4 also morally dominates contender no. 2, because he is ranked higher than contender no. 1 in the second and third discipline, and contender no. 1 is ranked higher than contender no. 2 in the first and second discipline.

Contender no. 1 morally dominates contender no. 5, because he is ranked higher in all the disciplines. However, contender no. 5 does *not* morally dominate contender no. 1. Indeed, contender no. 3 is the only one ranked lower than contender no. 5 in at least two disciplines, but contender no. 3 is not ranked higher than any other contender in at least two disciplines.

Sample grading tests:

`1ocen`: $n = 10$, $m = 90$, all standings in random order; queries for all pairs.

`2ocen`: $n = 1000$, $m = 1000$, same order in all disciplines, random queries.

`3ocen`: $n = 100\,000$, $m = 10$, standings for the first discipline: $1, 2, \dots, n$, standings for the second discipline: $n, n-1, \dots, 1$. Standings for the third discipline are randomly ordered. There are 10 random queries.

Grading

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

Subset	**Property**	**Score**
1	$n, m \le 100$	9
2	$n \le 300$, $m \le 100\,000$	10
3	$n \le 1000$, $m \le 1\,000\,000$	18
4	$n \le 100\,000$, $m \le 10$	27
5	$n \le 500\,000$, $m \le 1\,000\,000$	36