Bytehattan

Available memory: 128 MB.

Bytehattan is one of the islands in the capital of Byteland. The island is a witness to parades, outings and processions organized frequently. In fact these are so persistent, that street closing and serious traffic congestion result. Byteasar, employed at the town hall, has been appointed to monitor the city traffic.

The streets of Bytehattan form a regular $n \times n$ grid. Let us look at the map of Bytehattan as if it were grid coordinates: for each pair of integers x, y, such that $1 \leqslant x, y \leqslant n$, at point defined by its coordinates (x, y) there is an intersection. Each two street crossings 1 unit away are joined by a street measuring 1 unit.

Byteasar receives messages concerning street closures. Each message informs that a particular street will be closed from now on. After receiving such information concerning the closure of a given street, Byteasar should determine whether it would be still possible to commute between two intersections which are located at the ends of such a closed street, using roads which have not yet been closed. Help him and create a program helping him with his job.

Input

The first line of input contains two integers n and k ($2 \leqslant n \leqslant 1500$, $1 \leqslant k \leqslant 2n(n-1)$): the number of streets in Bytehattan and the number of messages concerning closed streets. Each of the following k lines contains information concerning the closure of one of the streets, information is provided in chronological order. Each of these lines consists of *two* streets described, one after the other. In practice exactly one of the streets becomes closed*. In case it is still possible to commute between two intersections which are located at the ends of such a closed street, described in the *previous line*, the first of these streets becomes closed. In case it is not possible, the second one is closed. The first closure, out of those k closures described in the input, applies to the first one out of the two streets listed. Each street can only be closed once.

The description of a given street is an integer pair a_i, b_i ($1 \leqslant a_i, b_i \leqslant n$) followed by a letter c_i ($c_i \in \{\mathtt{N}, \mathtt{E}\}$). Such a triple determines a street, with one of its ends is positioned at an intersection described by coordinates: (a_i, b_i). In case $c_i = \mathtt{N}$, the other end of the street is positioned at an intersection described by coordinates: $(a_i, b_i + 1)$. In case $c_i = \mathtt{E}$, the other end of the street is positioned at an intersection described by coordinates: $(a_i + 1, b_i)$. In case $c_i = \mathtt{N}$, then $b_i < n$, similarly if $c_i = \mathtt{E}$, then $a_i < n$.

Output

Exactly k lines should be contained in the output. In case after i-th street closure it is still possible to commute between the intersections on a closed street from input, in the i-th line of output there should be the word `TAK` (Polish for *yes*). Otherwise the i-th line should contain word `NIE` (Polish for *no*).

Example

For the input data:

```
3 4
2 1 E 1 2 N
2 1 N 1 1 N
3 1 N 2 1 N
2 2 N 1 1 N
```

the correct result is:

```
TAK
TAK
NIE
NIE
```

*The intention of the jury is that such an atypical input format would create the need for processing each street closure before starting the processing of subsequent street closures.

ORGANIZATOR Universitas Varsoviensis 25–27 X 2013 / amppz.mimuw.edu.pl
SPONSOR GŁÓWNY Bank Polski
MECENAS Fundacja
PARTNERZY SAMSUNG IBM intel facebook Software Mind