

Task: LOG

Logistics



XXII OI, Stage II, Day trial. Source file `log.*` Available memory: 256 MB.

10.02.2015

Byteasar owns a logistics company. Its clients often have the company transport vast amounts of goods, which do not fit in a single truck. In such cases, Byteasar sends a convoy. Sometimes, there are more drivers than trucks in a convoy, with the spare drivers riding along as passengers. We assume that each truck can carry arbitrarily many passengers. The drivers can choose to stop the convoy at any moment. Before resuming the journey, the drivers can embark any of the trucks, and change behind the wheel. There are neither lower nor upper bounds on the number of stops along the way.

To increase road traffic safety, the Byteotian Ministry of Transport has mandated limits on the time a truck driver can spend behind the wheel. Each driver, having passed their periodic psychophysical tests, receives an entry in their driving license that tells how many kilometers they can spend behind the wheel on a single trip.

Byteasar has asked you to write a program that would help him manage his team of n truck drivers. The program must handle the following two types of events:

- Update of the entry in the i -th driver's license. We assume that no driver has any entry in their driving license initially. Before receiving one, a driver cannot drive a truck.
- A query regarding the possibility of sending a convoy of c trucks on a route that is s kilometers long. As mentioned before, en route, the drivers can ride as passengers, and can freely swap. The shipments are handled sequentially, i.e., the next convoy sets out only once the previous one has returned.

Input

The first line of the standard input contains two integers, n and m ($1 \leq n, m \leq 1\,000\,000$), separated by a single space, specifying the number of drivers and events respectively. The m lines that follow specify the events.

In case of an entry update, the line contains the letter `U` and two integers, k and a ($1 \leq k \leq n$, $0 \leq a \leq 1\,000\,000\,000$), indicating that the k -th driver can drive (behind the wheel) a kilometers in one trip from this point on. In case of a query, the line contains the letter `Z` and two integers, c and s ($1 \leq c \leq n$, $1 \leq s \leq 1\,000\,000\,000$), indicating a query about sending a convoy of c trucks on a route of s kilometers.

In tests worth 33% of the total score, the additional condition $n, m \leq 1000$ holds. In tests worth 66% of the total score, the additional condition $a, s \leq 1\,000\,000$ holds.

Output

If the input contains z queries, then z lines should be printed to the standard output: the i -th one should contain the word `TAK` (Polish for *YES*) or `NIE` (Polish for *NO*), depending on the answer to the i -th input query.

Example

For the input data:

```
3 8
U 1 5
U 2 7
Z 2 6
U 3 1
Z 2 6
U 2 2
Z 2 6
Z 2 1
```

the correct result is:

```
NIE
TAK
NIE
TAK
```

Sample grading tests:

- 1ocen:** one driver, several entry updates and queries, with both positive and negative answers;
- 2ocen:** 1000 drivers, each can drive 1000 km; we want to send 1000 trucks on a 1 km route;
- 3ocen:** 500 000 drivers who can drive 1 km each; we are sending one truck on a 500 000 km route.