COCI '06 Contest 4 #6 Ispiti

It's exam time in Mirko's village. Everyone wants to pass the exam with as little effort as possible, which is not easy. Mirko realized that it would be best for him to find someone who knows more than him and learn from them. Everyone followed and now everyone is looking for someone to learn from.

We can model how well a student is prepared for the exam with two integers, A and B. The number A represents how well a student understands the subject, while the number B is proportional to the quantity of their knowledge.

As the head of the village, Mirko decided that a student will ask another student for help only if that student has both numbers greater than or equal to the first student's numbers (no student will ask someone who doesn't understand the subject as well as themselves or who knows less).

Additionally, students will try to minimize the difference in knowledge quantity (so that students don't bother those that are way better). If this choice is not unique, they will try to minimize the difference in understanding.

Mirko's village has recently become a very popular suburb and new students keep moving in (in time for the exam). With Mirko's strict rules, they get confused about Mirko's rules and don't know where to go). They decided to ask a programmer from a neighbouring village for help.

Input Specification

The first line of input contains an integer N $(1 \le N \le 200\,000)$, the number of queries and arrivals in the village. Each of the following N lines contains either:

- `D A B`, a student has moved in whose knowledge is A and B
- `P i`, the i-th student to move in wants to know whom to ask for help

The numbers A and B are between 1 and 2×10^9. No two students have both numbers equal.

Output Specification

For each query (`P i` line), output which student the i-th student should ask for help. The students are numbered in the order they moved into the village (starting from 1). If a student cannot be helped, output `NE`.

Sample Input 1

```
6
D 3 1
D 2 2
D 1 3
P 1
P 2
P 3
```

Sample Output 1

```
NE
NE
NE
```

Sample Input 2

```
6
D 8 8
D 2 4
D 5 6
P 2
D 6 2
P 4
```

Sample Output 2

```
3
1
```

Sample Input 3

```
7
D 5 2
D 5 3
P 1
D 7 1
D 8 7
P 3
P 2
```

Sample Output 3

```
2
4
4
```