Pawsome Cards

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In Meow's world, all doors are installed very highly advanced locks, they are integrated with different features, such as paw reading, purr detection, fur scan, biometric eye scanning and more! These digital locks are even used in vaults and high security facilities. But there's something even more a-paw-ling, they have yet to realize a major flaw in their locks!

Meow is a legendary spy known for its ability to break into any secret bases in its world. It was able to achieve this because it realized the flaw, which is that all doors and vaults in their world will always have a key-card access or length, x and width, y. Meow was able to create key cards that can access many different locks at once, but it has no control over the size of these cards. Meow's uncle, Meowtini is a leatherworker and wants to help Meow make a card holder for it to carry these cards on its missions, but the problem is that Meow has no idea how big of a card holder it needs to keep all these cards when it goes on a mission.

Your job is to find the dimensions of a card holder with length, l and width, w that can fit these cards. Your job is to create a system that accepts 2 types of queries:

1. $+$ x y – Meow creates a card of size $x \times y$
2. ? h w – Meow checks if the card holder with size h $\times$ w fits all the cards up to that moment

The card holder can hold cards when if it fulfils one of the two condition:

1. $x \leq h$ and $y \leq w$
2. $y \leq h$ and $x \leq w$

You are required to output an **INTEGER** that indicates if the wallet fits all the previous cards mentioned. (0 – doesn't fit, 1 – fits all cards)

Input

The first line contains an integer n ($2 \leq n \leq 10^4$) – the number of queries.

The next n lines follows these 2 types of query formats:

1. $+$ x y – Meow creates a card of size $x \times y$
2. ? h w – Meow checks if the card holder with size $h \times w$ fits all the cards up to that moment ($1 \leq h, w \leq 10^9$)

It is guaranteed that the queries start with type **1** before having any type **2** queries and that there will be at least one type **2** queries within the input data.

Output

For each query type **2**, print 1 if the card holder can fit all cards up to that point and 0 if it cannot.

Example

standard input	standard output
14 + 2 4 + 4 2 ? 1 10 + 4 5 ? 5 5 + 1 5 ? 100 10 ? 1 5 + 105 24 + 40 25 ? 200 23 ? 110 25 + 195230571 928276543 ? 1000000000 1000000000	0 1 1 0 0 1 1