

Bad Maths

time limit per test: 1 second (3 seconds for Python)
memory limit per test: 500 megabytes
input: standard input
output: standard output
points: 100 · problem code: badmaths

You are given equations over variables that each take a value of either 1 or -1 . Every equation has the form $a \times b = c$, where a and b are variables and c is either 1 or -1 .

Process the equations one by one. For each equation, decide whether it is consistent with the equations accepted so far: that is, whether there still exists an assignment of ± 1 to all variables that satisfies every previously accepted equation together with the current one.

- If it is consistent, output **Yes** and accept the equation.
- If it is inconsistent, output **No** and discard the equation (do not use it when checking later equations).

Input

The first line contains an integer n , the number of equations.

Each of the following n lines contains **a b c**, meaning $a \times b = c$, where a and b are variable names (nonempty strings of lowercase English letters) and c is either 1 or -1 .

Output

For each equation, output a line containing **Yes** if it is consistent with the equations accepted so far, or **No** otherwise.

Constraints

- $1 \leq n \leq 5 \times 10^5$
- $1 \leq |a|, |b| \leq 10$
- $c \in \{1, -1\}$

Example 1

Input

```
10
c c -1
a b 1
c c 1
d e 1
c e 1
c a 1
b e -1
b b -1
b b -1
e b 1
```

Output

```
No
Yes
Yes
Yes
Yes
Yes
No
No
No
Yes
```

Explanation

- $c \times c = -1$ is impossible for $c \in \{1, -1\}$, so the answer is **No** and the equation is discarded.
- $a \times b = 1$ is accepted.
- $c \times c = 1$ always holds, so it is accepted.
- The next three equations are all compatible with an assignment and are accepted. Together they force b and e to have the same sign, so $b \times e$ must equal 1.
- Therefore $b \times e = -1$ is inconsistent and discarded.
- Both copies of $b \times b = -1$ are impossible and discarded.
- $e \times b = 1$ matches the forced relation, so it is accepted.

Example 2

Input

```
20
c b 1
e c -1
b b 1
d f -1
a d -1
b c 1
d b 1
f e -1
d b 1
e d 1
c d 1
f a -1
e e 1
c b 1
d d 1
d e -1
b a 1
b a 1
d f 1
a f -1
```

Output

Yes
Yes
Yes
Yes
Yes
Yes
Yes
No
Yes
No
Yes
No
Yes
Yes
Yes
Yes
Yes
No
No
No
No