Problem A. Asynchronous Processor

Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given a program consisting of n instructions executed by a processor with a single integer register A, initially equal to 0. Each instruction is one of two types:

- "`+ v`" — performs $A := A + v$;
- "`= v`" — performs $A := v$.

The instructions in the program are numbered from 1 to n. Each instruction i initially has timestamp i.

Some instructions are marked as *asynchronous*. If instruction i is asynchronous, its timestamp can be changed to any **real** number greater than i.

After all timestamp adjustments, all timestamps must be distinct. The processor then executes the instructions in order of increasing timestamp.

Determine the number of distinct final values of A that can be obtained after all instructions have been executed, considering all possible choices of asynchronous instruction timestamps.

Input

The first line contains an integer n, denoting the number of instructions in the program ($1 \le n \le 2000$).

The i-th of the following n lines describes instruction i and contains three tokens. The first token is either '`+`' or '`=`', denoting the type of the instruction. The second token is an integer v, denoting the argument of the instruction ($1 \le v \le 500$). Finally, the third token is either "`async`" if the instruction is marked as asynchronous, or "`sync`" otherwise.

Output

Print the number of distinct final values A can take after executing the program.

Examples

standard input	standard output
3 + 1 sync = 2 async + 3 async	2
10 = 7 async + 3 async + 5 sync + 3 async = 1 sync + 9 async + 10 async + 1 sync + 3 async + 4 sync	30

Note

In the first test, the program execution starts with instruction 1 setting A to 1. Then, instructions 2 and 3 are executed in one of the two orders:

- if "`= 2`" is executed before "`+ 3`", A will be equal to 5;
- if "`+ 3`" is executed before "`= 2`", A will be equal to 2.

Thus, there are two possible values of A at the end: 5 and 2.