A. Bit++

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The classic programming language of Bitland is Bit++. This language is so peculiar and complicated.

The language is that peculiar as it has exactly one variable, called x. Also, there are two operations:

- Operation ++ increases the value of variable x by 1.
- Operation −− decreases the value of variable x by 1.

A statement in language Bit++ is a sequence, consisting of exactly one operation and one variable x. The statement is written without spaces, that is, it can only contain characters "+", "−", "X". Executing a statement means applying the operation it contains.

A programme in Bit++ is a sequence of statements, each of them needs to be executed. Executing a programme means executing all the statements it contains.

You're given a programme in language Bit++. The initial value of x is 0. Execute the programme and find its final value (the value of the variable when this programme is executed).

Input

The first line contains a single integer n $(1 \le n \le 150)$ — the number of statements in the programme.

Next n lines contain a statement each. Each statement contains exactly one operation (++ or −−) and exactly one variable x (denoted as letter «X»). Thus, there are no empty statements. The operation and the variable can be written in any order.

Output

Print a single integer — the final value of x.

Examples

input	Copy
1 ++X	

output	Copy
1	

input	Copy
2 X++ --X	

output	Copy
0	