Problem F. Postfix

Problem Description

Postfix notation is a mathematical notation in which operators follow their operands. It does not need any parentheses to indicate the order of operations when evaluating the values of the expression. It is found that using postfix notation may lead to faster calculations and can save more memory compared to other mathematical notations during evaluation.

Now, given a postfix expression, please write a program to calculate the value of it. Since the answer might be very large, please output the answer modulo $(10^9 + 7)$.

Input Format

Each input will contain one mathematical expression presented in the postfix notation. Each integer x in the expression would be a 32-bit integer, while operators op will be one of $\{+, *\}$. Two consecutive operands or operators will be seprated by a single space.

Output Format

Output an integer, the result of the expression.

Constraints

- $1 \leq x \leq 2^{31} - 1$.
- $op \in \{+, *\}$.
- All the numbers in inputs are integers.

Subtasks

1. (100 points) No additional constraints.

No.	Testdata Range	Time Limit (ms)	Memory Limit (KiB)
0	1	1000	262144
1	1-8	1000	262144

Samples

Sample Input 1

```
3 8 + 5 6 + *
```

Sample Output 1

```
121
```

The postfix expression can be solved by parsing the expression from the left, and do the calculation once we found two consequtive integers follow by a operator.

Takes this testcase for example. Starting from the left-hand side, first, we found `3 8 +` , which match the condition showing above, we do `3 + 8` and replace it back into the equation, then the equation become `11 5 6 + *`.

Keep doing the procedure that mentioned above, we can get the correct answer as showing below:

$$\begin{aligned}
&\texttt{3 8 + 5 6 + *} \\
\rightarrow\,&\texttt{11 5 6 + *} \\
\rightarrow\,&\texttt{11 11 *} \\
\rightarrow\,&\texttt{121}
\end{aligned}$$

Sample Input 2

```
2147483647 2147483647 *
```

Sample Output 2

```
850618742
```

Don't forget to output the answer modulo $(10^9 + 7)$.