

Problem

Math

Input File: MathIn.txt

Output File: MathOut.txt

Project File: Math

Mathematicians on the planet Earth, write math expressions using in-fixed notation. In this notation, math operators are written between the operands they operate on (e.g., $2 + 3$). On Mars, math strings are written in post-fixed form. In this notation, math operators are written after the two operands they operate on (e.g., $2\ 3\ +$).

The nice thing about post-fixed notation is that we don't need rules of precedence to decide what math operator should be evaluated first. For instance, in the in-fixed math string $6 + 4 / 2$, the rules of precedence dictate that we should divide before we add. Without these rules, there is an ambiguity in the expression. The same math expression written in post fixed notation is $4\ 2\ /\ 6\ +$.

Fortunately programmers who write translators are from Mars, and they translate math expressions from in-fixed to post-fixed notation before evaluating them. Thus, we need not worry about the rules of precedence at run time.

To evaluate a post-fixed string, we start at the left most character and examine characters until we find an operator. Once an operator is found, it is applied to the two operands immediately before it, and then the operand and the two operators in the post-fixed string are replaced with the result. Then we continue from this point, repeating the procedure. When we reach the end of the string, there will only be one item left in the string, the result. Thus the in-fixed string $5\ 6\ 2\ +\ 4\ /\ -$ is equivalent to the post-fixed string $5\ -\ (6\ +\ 2)\ /\ 4$, both of which evaluate to 3.

Inputs

The input file will contain math strings in post-fixed notation, one per line. Operands will consist of one digit. Operators and operands will be separated by one space. There will be no more than 80 characters in the math expressions.

Outputs

There will be one line of output for each math expression. The line will contain the value of the math expression.

Sample input

```
1 5 9 + 8 - +  
5 6 2 + 4 / -  
4 7 9 8 * + 2 + -
```

Sample Output

```
7  
3  
77
```