Stack

Time Limit : 1 sec, Memory Limit : 131072 KB

> Reverse Polish notation is a notation where every operator follows all of its operands. For example, an expression (1+2)*(5+4) in the conventional Polish notation can be represented as 1 2 + 5 4 + * in the Reverse Polish notation. One of advantages of the Reverse Polish notation is that it is parenthesis-free.

> Write a program which reads an expression in the Reverse Polish notation and prints the computational result.

> An expression in the Reverse Polish notation is calculated using a stack. To evaluate the expression, the program should read symbols in order. If the symbol is an operand, the corresponding value should be pushed into the stack. On the other hand, if the symbols is an operator, the program should pop two elements from the stack, perform the corresponding operations, then push the result in to the stack. The program should repeat this operations.

Input

> An expression is given in a line. Two consequtive symbols (operand or operator) are separated by a space character.

> You can assume that +, - and * are given as the operator and an operand is a positive integer less than 10^6

Output

> Print the computational result in a line.

Constraints

> 2 ≤ the number of operands in the expression ≤ 100
> 1 ≤ the number of operators in the expression ≤ 99
> $-1 \times 10^9 \leq$ values in the stack $\leq 10^9$

Sample Input 1

```
1 2 +
```

Sample Output 1

```
3
```

Sample Input 2

```
1 2 + 3 4 - *
```

Sample Output 2

```
-3
```

Notes

Template in C

Source: https://onlinejudge.u-aizu.ac.jp/problems/ALDS1_3_A