

## [PI043] - Eval

You are given a simple arithmetic expression consisting of integers, addition (+), and multiplication (\*). Unfortunately, the parentheses that originally defined the order of operations have been lost. To evaluate the expression correctly, you must follow the standard operator precedence rules: multiplication takes precedence over addition.

### Task

Your task is to compute the result of the expression by evaluating all multiplications before performing any additions.

### Input

The first line contains an integer  $n$  ( $1 \leq n \leq 9$ ), representing the number of numeric values in the expression.

The second line contains the expression, formatted as  $n$  numbers (each between 1 and 9, inclusive), separated by operators (+ or \*). Each token (number or operator) is separated by a single space.

### Output

Print a single line with the result of evaluating the expression according to operator precedence.

#### Example Input 1

```
3
5 + 2 * 3
```

#### Example Output 1

```
11
```

#### Example Input 2

```
1
2
```

#### Example Output 2

```
2
```

#### Example Input 3

```
3
1 + 2 + 3
```

#### Example Output 3

```
6
```

## [PI043] - Eval

É-lhe dada uma expressão aritmética simples composta por inteiros, adições (+) e multiplicações (\*). Infelizmente, os parêntesis que originalmente definiam a ordem das operações perderam-se. Para avaliar corretamente a expressão, deve seguir as regras padrão de precedência de operadores: a multiplicação tem precedência sobre a adição.

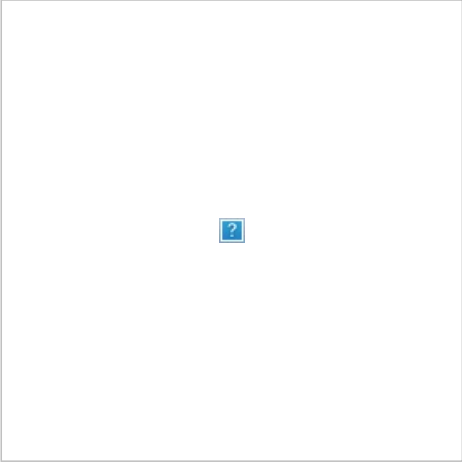
# Tarefa

A sua tarefa é calcular o resultado da expressão, avaliando todas as multiplicações antes de realizar quaisquer adições.

## Input

A primeira linha contém um inteiro  $n$  ( $1 \leq n \leq 9$ ), representando o número de valores numéricos na expressão.

A segunda linha contém a expressão, formatada com  $n$  números (cada um entre 1 e 9, inclusive), separados por operadores (+ ou \*). Cada elemento (número ou operador) está separado por um único espaço.



## Output

Escreva uma única linha com o resultado da avaliação da expressão de acordo com a precedência dos operadores.

### Example Input 1

```
3
5 + 2 * 3
```

### Example Output 1

```
11
```

### Example Input 2

```
1
2
```

### Example Output 2

```
2
```

### Example Input 3

```
3
1 + 2 + 3
```

### Example Output 3

```
6
```