

## split-into-tokens. Разделение выражения на лексемы

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Задано числовое выражение. Необходимо разбить его на лексемы и вывести каждую из них на новой строке. Гарантируется, что исходное выражение корректно. В выражении могут встречаться знаки сложения, вычитания, умножения и скобки, а также пробелы (пробелы не могут встречаться внутри чисел). Приоритет операций стандартный. Все числа в выражении целые.

### Формат входного файла

В первой строке вводится выражение. Его длина не превосходит 100 знаков. Выражение заканчивается точкой.

### Формат выходного файла

Выведите все встречающиеся лексемы выражения по порядку и ровно по одной на каждой строке. Никаких пробелов не должно быть выведено. Точка лексемой не считается.

### Примеры

стандартный ввод (или input.txt)	стандартный вывод (или output.txt)
1+(2*2 - 3).	1 + ( 2 * 2 - 3 )

## arithmetic-expr. Значение арифметического выражения

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Задано числовое выражение. Необходимо вычислить его значение или установить, что оно содержит ошибку. В выражении могут встречаться знаки сложения, вычитания, умножения, скобки и пробелы (пробелов внутри чисел быть не должно). Приоритет операций стандартный. Все числа в выражении целые.

### Формат входного файла

В первой строке вводится выражение. Его длина не превосходит 100 знаков. Выражение заканчивается точкой.

### Формат выходного файла

Выведите значение этого выражения, если оно не содержит ошибок, иначе выведите WRONG.

### Примеры

стандартный ввод (или input.txt)	стандартный вывод (или output.txt)
1+(2*2 - 3).	2
1+a+1.	WRONG
1 1 + 2.	WRONG

## logical-expr. Значение логического выражения

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Задано логическое выражение. Необходимо вычислить его значение. В выражении могут встречаться знаки ! (отрицание), & (логическое "и"), | (логическое "или"), ^ (XOR) и скобки. Самый высокий приоритет у отрицания, меньший приоритет у &, операции | и ^ имеют низкий приоритет (одинаковый) и вычисляются слева направо. В выражении встречаются только числа 0 и 1.

### Формат входного файла

В первой строке вводится выражение. Его длина не превосходит 100 знаков. Выражение заканчивается точкой.

### Формат выходного файла

Выведите значение этого выражения (0 или 1).

### Примеры

стандартный ввод (или input.txt)	стандартный вывод (или output.txt)
1 (0&0^1).	1

## new-year-expr. Новогоднее выражение

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Задано новогоднее выражение, заканчивающееся точкой. Необходимо посчитать его значение. В новогоднем выражении могут встречаться целые числа, знаки сложения, вычитания, умножения, возведения в степень (^), скобки, константы Ded Moroz, Moroz и Snegurochka, а также вызов функции Podarok(), которая принимает одно число на вход и возвращает его, увеличенное на 5, если оно было положительно, или возвращает его модуль, если оно было меньше или равно 0. Приоритет операций стандартный. Все числа в выражении целые. Также гарантируется, что при возведении в степень показатель степени положителен. Значения констант:

Ded Moroz	2013
Moroz	-30
Snegurochka	10

## Формат входного файла

В единственной строке вводится выражение. Его длина не превышает 200 знаков. Выражение заканчивается точкой.

## Формат выходного файла

Выведите значение этого выражения

## Примеры

стандартный ввод (или input.txt)	стандартный вывод (или output.txt)
Podarok(Moroz-Ded Moroz)*2.	4086
Snegurochka-30.	-20
3^4^2.	43046721

## Примечание

Обратите внимание, что операция возведения в степень вычисляется справа налево, то есть в третьем примере  $3^4^2$  вычисляется как  $3^{(4^2)}$

## smilies. Смайлики

Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Напишите программу, которая посчитает количество смайликов в заданном тексте.

Смайликом будем считать последовательность символов, удовлетворяющую условиям:

- первым символом является либо ; (точка с запятой) либо : (двоеточие) ровно один раз;
- далее может идти символ - (минус) сколько угодно раз (в том числе символ минус может идти ноль раз);
- в конце обязательно идет некоторое количество (не меньше одной) одинаковых скобок из следующего набора: (, ), [, ];
- внутри смайлика не может встречаться никаких других символов.

Например, нижеприведенные последовательности являются смайликами:

```
:)
;-----[[[[[[[[
```

в то время как эти последовательности смайликами не являются (хотя некоторые из них содержат смайлики):

```
:-)
;-
-)
::-(
:-()
```

## Формат входного файла

Вводится одна строка текста, которая может содержать маленькие латинские буквы, пробелы, символы, которые могут встречаться в смайликах. Длина строки не превышает 100 000 символов.

## Формат выходного файла

Выведите одно число — количество смайликов, которые встречаются в тексте.

## Примеры

стандартный ввод (или input.txt)	стандартный вывод (или output.txt)
:);---[[[[[[	2
:-)];--;	1
)-(--:--	0
hello :-)	1