

Задача А. Значение выражения

Имя входного файла: `expr.in`
Имя выходного файла: `expr.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Задано числовое выражение, заканчивающееся точкой. Необходимо посчитать его значение или сказать, что оно содержит ошибку. В выражении могут встречаться знаки сложения, вычитания, умножения и скобки. Приоритет операций стандартный. Все числа в выражении целые и принадлежат диапазону `LongInt`. Также гарантируется что все промежуточные вычисления умецаются в этот тип.

Формат входного файла

Первая строка входного файла содержит заданное выражение. Его длина не превосходит 100 знаков. Гарантируется, что выражение заканчивается точкой.

Формат выходного файла

Выведите в выходной файл значение этого выражения или слово `WRONG`, если значение не определено.

Пример

<code>expr.in</code>	<code>expr.out</code>
<code>1+(2*2-3).</code>	<code>2</code>
<code>1+a+1.</code>	<code>WRONG</code>

на новой строке. Гарантируется, что исходное выражение корректно.

В выражении могут встречаться знаки сложения, вычитания, умножения и скобки. Приоритет операций стандартный. Все числа в выражении целые и принадлежат диапазону `LongInt`.

Формат входного файла

Первая строка входного файла содержит заданное выражение. Его длина не превосходит 100 знаков. Гарантируется, что выражение заканчивается точкой.

Формат выходного файла

Выведите все встречающиеся лексемы выражения по порядку и ровно по одной на каждой строке.

Пример

<code>lexem.in</code>	<code>lexem.out</code>
<code>1+(2*2-3).</code>	<code>1</code>
	<code>+</code>
	<code>(</code>
	<code>2</code>
	<code>*</code>
	<code>2</code>
	<code>-</code>
	<code>3</code>
	<code>)</code>

Задача В. Регулярное выражение

Имя входного файла: `regex.in`
Имя выходного файла: `regex.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Назовём образцами те строчки, внутри которых могут помимо букв встречаться ещё и специальные символы - звёздочки. Образец можно превратить в обычную строчку, если каждую звёздочку заменить на какую-то, возможно пустую, строчку из букв. В таком случае мы говорим, что получившаяся строчка соответствует образцу. По данным строчке S и образцу T определите, соответствует ли S данному образцу.

Формат входного файла

В первой строке записаны числа n и m - длины S и T соответственно ($1 \leq n, m \leq 100$). Во второй строчке записана S , в третьей - T . В записи S могут встречаться только маленькие латинские буквы, а в записи T ещё и звёздочки.

Формат выходного файла

Выведите в выходной файл `YES`, если S соответствует T , и `NO` в противном случае.

Примеры

<code>regex.in</code>	<code>regex.out</code>
<code>3 3</code> <code>abc</code> <code>a*c</code>	<code>YES</code>
<code>7 5</code> <code>abacaba</code> <code>a*d*a</code>	<code>NO</code>

Задача С. Разделение выражения на лексемы

Имя входного файла: `lexem.in`
Имя выходного файла: `lexem.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Задано числовое выражение, заканчивающееся точкой. Необходимо разбить его на лексемы и вывести каждую

Задача D. Значение логического выражения

Имя входного файла: `logexpr.in`
Имя выходного файла: `logexpr.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Задано логическое выражение, заканчивающееся точкой. Необходимо посчитать его значение или сказать, что оно содержит ошибку. В выражении могут встречаться знаки ! (отрицание), & (логическое "и"), | (логическое "или"), ^ (XOR) и скобки. Самый большой приоритет у отрицания, чуть меньше у &, у оставшихся операций приоритет одинаково маленький. Все числа в выражении либо 0, либо 1.

Формат входного файла

Первая строка входного файла содержит заданное выражение. Его длина не превосходит 100 знаков. Гарантируется, что выражение заканчивается точкой.

Формат выходного файла

Выведите в выходной файл значение этого выражения (0 или 1).

Пример

<code>logexpr.in</code>	<code>logexpr.out</code>
<code>1 (0&0~1).</code>	<code>1</code>