

Задача А. Разделение выражения на лексемы

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

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

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

Формат выходных данных

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

Примеры

<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. Значение выражения

Имя входного файла: `expr.in`
Имя выходного файла: `expr.out`
Ограничение по времени: 2 секунды
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 уместятся в этот тип.

Формат входных данных

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

Формат выходных данных

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

Примеры

<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>

Задача С. Новогоднее выражение

Имя входного файла: `expr2.in`
Имя выходного файла: `expr2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Задано новогоднее выражение, заканчивающееся точкой. Необходимо посчитать его значение или сказать, что оно содержит ошибку. В новогоднем выражении могут встречаться знаки сложения, вычитания, умножения и 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 либо равно 0. Приоритет операций стандартный. Все числа в выражении целые и принадлежат диапазону `LongInt`. Также гарантируется, что все промежуточные вычисления уместятся в этот тип.

Формат входных данных

Первая строка входного файла содержит заданное выражение. Его длина не превосходит 200 знаков. Гарантируется, что выражение заканчивается точкой.

Значения констант:

Ded Moroz	2018
Moroz	-30
Snegurochka	10

Формат выходных данных

Выведите в выходной файл значение этого новогоднего выражения или слово «**WRONG**», если значение не определено.

Примеры

<code>expr2.in</code>	<code>expr2.out</code>
<code>Podarok(Moroz-Ded Moroz)*2.</code>	<code>4096</code>
<code>Snegurochka-30.</code>	<code>-20</code>

Задача D. R2-D2

Имя входного файла: r2d2.in
Имя выходного файла: r2d2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Во время очередного боя на очередном космическом корабле у R2-D2 заклинило некоторые платы. Случилось это в самый неподходящий момент: он попался во вражеское силовое поле. R2-D2 считает, что у поля координаты $[-N..N] \times [-N..N]$. На границе силового поля дроида может вытащить и починить Люк Скайуокер, но до границы ещё нужно пройти.

В начальный момент времени R2-D2 стоит в точке $(0, 0)$ и направлен в сторону точки $(1, 0)$.

У R2-D2 остался доступ лишь к одному логическому выражению и регистрам, хранящим значения переменных. Выражение содержит операции NOT, AND, OR (NOT имеет наибольший приоритет, OR — наименьший), скобки, константы TRUE и FALSE, а переменные (регистры) находятся в диапазоне 'A', ..., 'Z'.

Изначально, все регистры дроида имеют значение FALSE. R2-D2 движется по прямой, пока не достигнет развилки — точки, в которой у него клинит контакт. После этого дроид вычисляет значение выражения и поворачивает направо, если оно истинно, и налево в противном случае. Однако при попадании в некоторые точки поля один из регистров дроида меняет свое значение на противоположное.

Вас попросили найти маршрут R2-D2 до момента, когда он выйдет за пределы поля.

Формат входных данных

В первой строке файла содержится логическое выражение. Длина выражения не превышает 250 символов.

Вторая строка содержит три целых числа N, M, K ($1 \leq N \leq 100, 1 \leq M \leq 100, 1 \leq K \leq 100$), где N — размер поля, M — количество развилок, K — количество точек перемены регистров.

В следующих M строках содержатся два целых числа X, Y — координаты развилки.

Каждая из следующих K строк содержит два целых числа X, Y и символ C — координаты точки перемены регистра и название регистра, значение которого инвертируется.

Вы можете предполагать, что в точке $(0, 0)$ нет развилки, что в каждой точке поля находится не более одного объекта (развилки или точки перемены регистра). Также гарантируется, что R2-D2 в какой-то момент выйдет за пределы поля.

Формат выходных данных

Выведите в выходной файл путь R2-D2 — каждая пара координат в отдельной строке.

Примеры

r2d2.in	r2d2.out
NOT((A OR NOT B) AND (A OR B)) OR NOT	0 0
(A AND NOT B OR TRUE)	1 0
1 5 2	1 -1
1 0	0 -1
1 1	-1 -1
1 -1	-1 0
-1 -1	-1 1
-1 1	0 1
0 1 A	1 1
-1 0 D	