

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

lexem.in	lexem.out
1+(2*2-3).	1
	+
	(
	2
	*
	2
	-
	3
)

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:

<code>Ded Moroz</code>	2018
<code>Moroz</code>	-30
<code>Snegurochka</code>	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>	4096
<code>Snegurochka-30.</code>	-20

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	

Задача Е. Дерево разбора

Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайт

Арифметические выражения, использующие сложение, вычитание, умножение, деление и возведение в степень определяются следующей грамматикой:

Сложение, вычитание, умножение и деление левоассоциативны, а возведение в степень правоассоциативно.

Для арифметического выражения определено его *дерево разбора*. Это двоичное дерево, в котором внутренние узлы соответствуют бинарным операциям, а листья соответствуют переменным. Дерево строится рекурсивно.

- Дерево для переменной — это дерево из одной вершины, в которой записана эта переменная.
- Дерево для элемента, являющегося выражением в скобках — это дерево для самого выражения.
- Дерево для множителя, являющегося элементом — это дерево для этого элемента. Дерево для множителя вида «элемент e , возведённый в f » — это дерево, в котором в корне записана операция '^', левое поддерево корня — дерево для элемента e , правое поддерево корня — дерево для множителя f .
- Деревья для множителя и слагаемого определяется аналогично, с тем лишь различием, что соответствующие операции лево-ассоциативные.

Вам дано арифметическое выражение, выведите его дерево разбора.

Формат входных данных

Во входном файле содержится корректное арифметическое выражение, состоящее не более, чем из 400 символов

Формат выходных данных

Во входной файл выведите дерево разбора.

Дерево разбора для переменной должно быть размера 1×1 и содержать эту переменную.

Дерево, в корне которого записана операция, с поддеревьями T_1 и T_2 , которые имеют размеры $h_1 \times w_1$ и $h_2 \times w_2$ соответственно, должно быть размера $(\max\{h_1, h_2\} + 2) \times (w_1 + w_2 + 5)$.

Подробнее о формате вывода можно узнать, изучив пример выходного файла (см. ниже). Следует использовать следующие вспомогательные символы: минус '-' (код ASCII 45), точка '.' (код ASCII 46), вертикальная черта '|' (код ASCII 124), квадратные скобки '[' и ']' (коды ASCII 91 and 93).

Примеры

стандартный ввод	стандартный вывод
$(a+b+c)*(d-a)$	<pre> .----[*]----. .----[+]-. .-[]-. .-[+]-. c d a a b </pre>

Задача F. Кьяк-у-пакал и иероглифы Майя

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Когда археологи проводили раскопки в древних городах Майя, они обнаружили множество непонятных иероглифов. Пример иероглифа показан справа, он обозначает имя Кьяк-у-пакал, это имя военного и религиозного лидера в древнем городе Майя Чичен-Итца (см. А.В.Восс, Г. Дж.Кремер Кьяк-у-пакал, Хун-пик-токь и Коком). Этот иероглиф можно увидеть во многих местах древнего города.

Вообще говоря, иероглифы Майя не являются иероглифами в прямом смысле этого слова, а, скорее, являются композицией отдельных глифов. Все известные глифы занумерованы целыми числами от 1 до 9999. Учеными был разработан специальный язык, с помощью которого можно представлять иероглиф в виде обычного текста. Например, иероглиф Кьяк-у-пакал кодируется как «((669:604).(586:(27:(534.534))))».

Приведем формальную грамматику этого языка:

```
<inscription> <glyph id>|'|('<block>'.<horizontal group>')'|'|('<block>':<vertical group>')'  
<horizontal group> <block>['.<horizontal group>']'  
<vertical group> <block>[':<vertical group>']  
<block> <inscription>|'|['<inscription>']'
```

Код иероглифа описывает процесс его составления. Глифы комбинируются горизонтально и вертикально (с помощью ':' и '.') в блоки, которые, в свою очередь, комбинируются во все большие и большие блоки, до тех пор, пока не будет достигнута нужная конфигурация. Как обычно, в процессе реализации забыли о важной части — обратном восстановлении обычного текста в иероглиф. Это предстоит сделать вам.

Формат входных данных

Единственная строка входного файла содержит строку, описывающую иероглиф Майя в виде обычного текста. Длина строки не превышает 255 символов. Строка не содержит пробелов.

Формат выходных данных

Выведите текст, составленный из символов '+', '-', '|', ' ' (ASCII коды 43, 45, 124, 32), '0'..'9' и переводов строки. Все блоки должны иметь одинаковый размер, кроме блоков, окруженных квадратными скобками (максимум в группе содержится один такой блок), которые должны быть вдвое шире (или выше, если группа вертикальная). Номер глифа (glyph id) с одним пробелом перед ним и одним пробелом после, должен быть помещен в левый верхний угол блока. Вывод должен быть как можно короче. Гарантируется, что для всех тестов существует изображение, содержащее максимум 100 000 байт.

Примеры

стандартный ввод	стандартный вывод
((669:604).(586:(27:(534.534))))	+-----+-----+ 669 586 +-----+-----+ 604 27 +-----+-----+ 534 534 +-----+-----+-----+-----+

Задача G. Числа Каталана

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Эндрю любит числа Каталана, а ещё больше Эндрю любит шутки.

Являясь достаточно опытным автором задач, Эндрю готовит достаточно много контестов для различных лагерей и сборов. В каждом контесте он даёт задачу, в которой входные данные и выходные данные состоят из одного числа, а ответами для 0, 1, 2, 3, 4 и 5 являются 1, 1, 2, 5, 14 и 42 соответственно. Однако ответы на большие тесты не являются соответствующими числами Каталана.

Эндрю уже подготовил так много контестов, что идеи для хороших задач с таким свойством начали заканчиваться, поэтому он решил автоматизировать процесс создания таких задач. В качестве хорошего источника таких задач Эндрю рассматривает проблемы подсчёта строк специальной длины в некотором детерминированном конечном автомате. Эндрю выбрал число k — желаемый ответ на задачу для теста 6 и теперь хочет найти детерминированный конечный ответ, который приводит к ответам 1, 1, 2, 5, 14, 42, k для входных данных 0, 1, 2, 3, 4, 5, 6.

По заданному k найдите и выведите детерминированный конечный автомат с таким свойством и с алфавитом не превосходящим 20, среди таких автоматов выберите тот, в котором число состояний минимально.

Формат входных данных

Входной файл состоит из нескольких тестов, разделённых переводами строк, каждый тест представляет из себя одно число k ($120 \leq k \leq 140$).

Входной файл завершается строкой с $k = 0$.

Формат выходных данных

Для каждого теста выведите автомат, являющийся ответом в следующем виде:

В первой строке выведите n — число состояний и s — размер алфавита ($1 \leq s \leq 20$).

Обратите внимание, что вам требуется минимизировать n и не требуется минимизировать s .

Пронумеруйте буквы от 1 до s и состояния от 1 до n , где состояние 1 является начальным состоянием автомата.

Во второй строке выведите n нулей или единиц, определяющих является ли соответствующее состояние терминальным.

В i -ой из последующих n строк выведите s чисел: куда ведёт переход из i -го состояния по соответствующему символу, если из i -го состояния нет перехода по соответствующему символу выведите 0.

Примеры

стандартный ввод	стандартный вывод
131	3 4
0	1 0 0
	1 2 0 0
	1 2 2 3
	2 3 3 0

Замечание

Если обозначить алфавит в примере как $\Sigma = \{a, b, c, d\}$, то пятью словами длины 3, которые принимает автомат, являются “aaa”, “aba”, “baa”, “bba” и “bca”.