Задача A. 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 B. 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 умещаются в этот тип. Унарный плюс и унарный минус в выражении встречаться не могут, как и два знака подряд.

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

expr.in	expr.out
`1+(2*2-3).`	`2`
`1+a+1.`	`WRONG`

Задача C. 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 умещаются 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	2020
Moroz	−30
Snegurochka	10

Формат выходных данных

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

Примеры

expr2.in	expr2.out
`Podarok(Moroz-Ded Moroz)*2.`	`4100`
`Snegurochka-30.`	`-20`

Задача D. Компактная запись

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

Для компактной записи длинных строк из заглавных латинских букв будем использовать следующую сокращенную запись: строка N(S), где N — натуральное число, S — строка, означают "повторить строку S N раз". Если строка S состоит из одной буквы, то скобки могут опускаться. Например, строка `AABAAB` может быть записана как `2(2AB)`.

Напишите программу, которая по сокращенной записи восстанавливает исходную строку.

Формат входных данных

Во входном файле задана строка в компактном формате.

Формат выходных данных

Выведите в выходной файл исходную строку. Гарантируется, что ее длина не превышает 1000 символов.

Примеры

compact.in	compact.out
2(2AB)	AABAAB
10A10B	AAAAAAAAAABBBBBBBBBB

Задача E. 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 правоасоциативно.

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 дере 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. 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)	

Задача F. Lawful Good

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

Рассмотрим язык, похожий на Си. В нём есть следующие простые типы:

- `char` — занимает один байт
- `short` — занимает два байта
- `int` — занимает четыре байта
- `long` — занимает восемь байт

В языке есть оператор `sizeof`, который позволяет узнать размер любого типа в байтах. К примеру, `sizeof(int)` равен четырём.

В языке есть оператор `alignof`, который позволяет узнать **выравнивание** любого типа в байтах. Адрес переменной какого-то типа `T` в памяти должен делиться на `alignof(T)`. `sizeof(T)` всегда делится на `alignof(T)`. `alignof(T)` всегда является неотрицательной целой степенью двойки. Для простых типов, `alignof(T) == sizeof(T)`.

В языке есть массивы фиксированной длины, состоящие из элементов одного типа. Массив из `n` элементов, каждый типа `T` обозначается как `T[n]`. `sizeof(T[n])` равен `sizeof(T) * n`. К примеру, `sizeof(short[13])` равен 26, так как размер типа `short` — два байта, а в массиве 13 элементов. `alignof(T[n])` равен `alignof(T)`.

В языке есть структуры — композитные типы, позволяющие объединять фиксированное количество переменных (полей) разных типов в одну. Пусть в структуре $n > 0$ полей $f_1, \ldots, f_n$ типов $T_1, \ldots, T_n$. Пусть эта структура лежит в памяти по адресу a. Тогда должны выполняться следующие дополнительные условия:

- Адрес f_1 равен a.
- Для $k = 2, \ldots, n$, Адрес f_k больше адреса f_{k-1}.
- Поля не могут пересекаться
- Как и для самой структуры, так и для всех её полей должны выполнятся стандартные правила выравнивания.
- Выравнивание структуры — максимум из выравниваний её полей.
- Размер структуры не меньше суммы размеров её полей.
- Размер структуры — минимальный из размеров, удовлетворяющий всем условиям.

Вам дано описание структуры `X`, состоящей из простых типов и одномерных массивов простых типов. Найдите `sizeof(X)` и `alignof(X)`.

Формат входных данных

В первой строке входного файла записано натуральное число n — количество полей структуры `X`. $1 \leqslant n \leqslant 10^5$.

В последующих n строках записаны типы полей. Каждый тип — это либо простой тип, либо одномерный массив из простых типов.

Гарантируется, что размер структуры `X` не превосходит одного гигабайта.

Формат выходных данных

На первой строке выведите одно целое число: `sizeof(X)`.

На второй строке выведите одно целое число: `alignof(X)`.

Примеры

стандартный ввод	стандартный вывод
3 char[2] int char[2]	12 4
2 char[4] int	8 4

Задача G. Chaotic Evil

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

Рассмотрим язык, чуть более похожий на Си, чем в прошлый раз. В нём есть следующие простые типы:

Имя типа	Размер
`bool`	1 байт
`char` или `signed char`	1 байт
`unsigned char`	1 байт
`short`, `signed short`, `short int` или `signed short int`	2 байта
`unsigned short` или `unsigned short int`	2 байта
`int`, `signed` или `signed int`	4 байта
`unsigned` или `unsigned int`	4 байта
`long`, `signed long`, `long int` или `signed long int`	8 байт
`unsigned long` или `unsigned long int`	8 байт
`long long`, `signed long long`, `long long int` или `signed long long int`	8 байт
`unsigned long long` или `unsigned long long int`	8 байт

В языке есть оператор `sizeof`, который позволяет узнать размер любого типа в байтах. К примеру, `sizeof(int)` равен четырём.

В языке есть оператор `alignof`, который позволяет узнать **выравнивание** любого типа в байтах. Адрес переменной какого-то типа `T` в памяти должен делиться на `alignof(T)`. `sizeof(T)` всегда делится на `alignof(T)`. `alignof(T)` всегда является неотрицательной целой степенью двойки. Для простых типов, `alignof(T) == sizeof(T)`.

В языке есть массивы фиксированной длины, состоящие из элементов одного типа. Массив из `n` элементов, каждый типа `T` обозначается как `T[n]`. `sizeof(T[n])` равен `sizeof(T) * n`. К примеру, `sizeof(short[13])` равен 26, так как размер типа `short` — два байта, а в массиве 13 элементов. `alignof(T[n])` равен `alignof(T)`. Поддержки многомерных массивов в языке нет.

В языке есть структуры — композитные типы, позволяющие объединять фиксированное количество переменных (полей) разных типов в одну. Пусть в структуре $n > 0$ полей $f_1, \ldots, f_n$ типов $T_1, \ldots, T_n$. Пусть эта структура лежит в памяти по адресу a. Тогда должны выполняться следующие дополнительные условия:

- Адрес f_1 равен a.
- Для $k = 2, \ldots, n$, Адрес f_k больше адреса f_{k-1}.
- Поля не могут пересекаться
- Как и для самой структуры, так и для всех её полей должны выполнятся стандартные правила выравнивания.
- Выравнивание структуры — максимум из выравниваний её полей.
- Размер структуры не меньше суммы размеров её полей.
- Размер структуры — минимальный из размеров, удовлетворяющий всем условиям.

Вам предлагается написать программу, вычисляющую `sizeof` и `alignof` для произвольных типов.

Формат входных данных

Во входном файле записаны команды, `typedef`, `sizeof` или `alignof`.

Команда `typedef` объявляет новый тип. Например, `typedef eightbytes unsigned char[8]` объявляет новый тип `eightbytes`, который представляет собой массив из восьми `unsigned char`. `typedef` может также объявлять структуры. Смотрите примеры. Гарантируется, что имя нового типа — непустая строка из латинских букв длиной не более 32 символов, кроме `bool`, `char`, `signed`, `unsigned`, `short`, `int`, `long`, `struct`, `typedef`, `sizeof`, `alignof`

Гарантируется, что объявления новых типов имеют уникальные имена.

Команды `sizeof` и `alignof` печатают на экран на новой строке размер и выравнивание типа соответственно. Например, `sizeof unsigned char[8]` напечатает на экран `8`.

Гарантируется, что размер входных данных по объёму не превосходит одного мегабайта. Размер каждого используемого типа не превосходит одного эксабайта.

Формат выходных данных

Для каждой команды `sizeof` или `alignof` в отдельной строке напечатайте результат выполнения соответствующего оператора.

Примеры

стандартный ввод	стандартный вывод
typedef eightbytes unsigned char[8] sizeof eightbytes alignof eightbytes	8 1

typedef verylong struct {
long[2];
unsigned short int[4];
}
sizeof verylong
alignof verylong
	24
	8

typedef verylong struct {
long[2];
unsigned short int[4];
}
typedef evenlonger struct {
verylong[4];
}
sizeof evenlonger
alignof evenlonger
typedef arr evenlonger[123]
sizeof arr
alignof arr
	96
	8
	11808
	8