

Задача А. Магическое выражение

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

Совсем недавно вас приняли в Орден пиромантов — магов огня. И буквально на следующий день между вашим Орденом и некромантами — магами, вызывающими мертвецов — началась война. Когда между магами идёт война, заклинания летят во все стороны. Однако нужно не только нападать, но и защищаться. Собственно, этим вы и займётесь.

Вражеские маги наслали на вас атакующее заклинание, прокричав его. *Атакующее заклинание* можно записать в виде арифметического выражения, состоящего из целых чисел от 0 до 9, знаков бинарных операций «+», «-» и «*», скобок «(» и «)», а также символов «x», обозначающих неизвестную величину.

Чтобы отбить вражеское атакующее заклинание, необходимо после него прокричать своё, защитное. *Защитное заклинание* можно записать в виде строки, которую можно дописать справа к любому атакующему заклинанию и вновь получить арифметическое выражение. После этого два заклинания сольются в единое целое. Если значение выражения, получившееся в итоге, будет чётным при любом возможном значении x , тогда атакующее заклинание не подействует, в противном же случае вам будет больно.

Величина x может принимать любое целочисленное значение от 0 до 9 и зависит от погодных условий на поле боя. Если x встречается в выражении больше одного раза, значение этой величины во всех позициях будет одним и тем же.

Вычисление значения выражения происходит по стандартным математическим правилам. Операции выполняются слева направо. Сначала вычисляется то, что в скобках, далее производится умножение, а затем выполняются сложение и вычитание. Унарные операции в выражениях не допускаются: например, строки «-2» и «+2» не являются атакующими заклинаниями.

Ваша задача — найти такое защитное заклинание, чтобы значение выражения, получившегося в итоге, было чётным независимо от значения неизвестной x .

Формат входных данных

В первой строке ввода содержится атакующее заклинание — арифметическое выражение длины не более 500, записанное без пробелов. Входные данные завершаются символом перевода строки. Гарантируется, что заданное выражение корректно и может содержать только целые числа от 0 до 9, знаки «*», «+», «-», «(», «)», а также символы «x».

Формат выходных данных

Выведите защитное заклинание: одну строку, не содержащую пробелов и состоящую не более, чем из 1000 символов. При дописывании этой строки справа к заданному атакующему заклинанию должно получаться арифметическое выражение, значение которого чётно при любом целочисленном значении x от 0 до 9. Ваша строка может быть пустой, но даже в этом случае необходимо создать выходной файл. Если возможных ответов несколько, разрешается вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
$x+2*x$	$+x*3*5$
$4+x$	$*2$
$(x-3)*x$	

Задача В. $P1 \mid p_i = 1 \mid \sum U_i$

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Задача $P1 \mid p_i = 1 \mid \sum U_i$. Дана одна машина и n работ. Для каждой работы известен её дедлайн d_i . Каждая работа выполняется за единицу времени. Нужно минимизировать количество невыполненных работ.

Формат входных данных

В первой строке дано семь целых чисел: n ($2 \leq n \leq 30\,000\,000$), d_1, d_2, A, B, C, D ($0 \leq d_1, d_2, A, B, C \leq 1\,000\,000\,000$, $1 \leq D \leq 1\,000\,000\,000$), где n — количество работ, d_1, d_2 — дедлайны первой и второй работ соответственно, A, B, C, D — коэффициенты для расчета дедлайнов последующих работ. Для $i > 2$ дедлайн рассчитывается по формуле $d_i = (A \cdot d_{i-2} + B \cdot d_{i-1} + C) \% D$.

Формат выходных данных

В первой строке выведите единственное целое число — максимальное количество выполненных работ.

Примеры

стандартный ввод	стандартный вывод
5 1 1 3 1 2 10	2

Задача С. Красное и черное

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

Вы играете в игру по следующим правилам: вы берете колоду из r красных и b черных карт, перемешиваете ее, после чего открываете карты по одной. Когда вы открываете красную карту, вы получаете очко, когда открываете черную — теряете. (В процессе игры ваш баланс может стать отрицательным.) В любой момент (в том числе, еще не открыв первую карту) вы можете прекратить игру, и тогда ваш текущий счет станет окончательным.

Какого математического ожидания вашего окончательного счета, если колода было перемешана идеально (все перестановки карт равновероятны), а вы играете оптимально?

Формат входных данных

Во входном файле содержатся два числа r и b ($0 \leq r, b \leq 5000$).

Формат выходных данных

Выведите математическое ожидание окончательного счета игры при оптимальной стратегии с абсолютной или относительной погрешностью не более 10^{-9} .

стандартный ввод	стандартный вывод
0 7	0.0000000000000000
4 0	4.0000000000000000
5 1	4.166666666666667

Задача D. 0*0

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

Юный ЛКШонок Серёженька знает, что некоторые множества строк можно описывать с помощью регулярных выражений, но почти не знает как.

Пока он умеет работать только с бинарными строками (то есть состоящими только из нулей и единиц), а из синтаксиса регулярных выражений знает лишь два правила: «0*» означает «строка, состоящая только из нулей (возможно, пустая)», а «1*» означает «строка, состоящая только из единиц (возможно, пустая)».

Например, регулярное выражение «01*00*» задает множество строк, которые начинаются на «0», затем имеют ноль или более «1», затем «0», затем ноль или более «0». Кратчайшая строка, принадлежащая этому множеству, — это «00».

Вычислите, сколько строк, задаваемых данным регулярным выражением R имеют длину не более L .

Формат входных данных

Во входном файле содержится регулярное выражение R , содержащее от 1 до 50 символов, состоящее только из символов «0», «1» и «*», причем перед каждой «*» находится «0» или «1»; после которого содержится число L ($1 \leq L \leq 50$) — максимальная длина строки.

Формат выходных данных

Выведите количество строк длины не более L , удовлетворяющих регулярному выражению R .

стандартный ввод	стандартный вывод
0* 5	6
01*1*1* 3	3
0*0*011*0*0 3	1
0*0*011*0*0 4	4