

Задача А. Сбалансируй-ка!

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

Лидеру команды «Отбой» на День Рождения подарили подвешенное бинарное дерево. Однако, ему не понравилось, что дерево было несбалансированным. Теперь он хочет удалить минимальное количество вершин в дереве, чтобы оно стало сбалансированным. Перед тем как удалить вершину из дерева, он обязан удалить все вершины из её поддерева. Напомним, что дерево является сбалансированным тогда и только тогда, когда для любой вершины высота её левого и правого поддеревьев отличается не более чем на 1 (высота пустого дерева равна нулю, а высота дерева из одной вершины — единице). Корнем дерева является вершина 1.

Формат входных данных

В первой строке входного файла задано целое число n — количество вершин в дереве ($1 \leq n \leq 1111$). В следующих n строках заданы по два целых числа $left_i$ и $right_i$ — номера левого и правого ребёнка вершины соответственно или 0, если этого ребёнка не существует.

Формат выходных данных

В единственной строке выходного выведите одно число — искомое минимальное количество удаляемых вершин.

Примеры

<code>balance.in</code>	<code>balance.out</code>
6 2 3 0 0 4 5 0 6 0 0 0 0	1
3 0 2 0 3 0 0	1

Задача В. Собеседование в «Отбой»

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

Вы проходите собеседование в легендарную команду «Отбой». Вы успешно отвечали на

все вопросы, и теперь от главного приза — членства в команде «Отбой», дающего доступ к безлимитным запасам кефирчика, вас отделяет последняя задача — посчитать количество чисел на отрезке с l по r , сумма цифр которых кратна числу k . Более формально — вам нужно посчитать количество чисел i ($l \leq i \leq r$), у которых сумма цифр кратна k .

Формат входных данных

В единственной строке входного файла заданы три числа l, r, k ($1 \leq l \leq r \leq 10^{18}, 1 \leq k \leq 1000$).

Формат выходных данных

В единственную строку выходного файла выведите ответ на задачу.

Примеры

<code>coolnumbers.in</code>	<code>coolnumbers.out</code>
1 10 1	10
1 10 2	4

Задача С. Шоколадка

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

Команда «Отбой» участвует в очередном марафоне по «Угадай мелодию. Rock version». Чтобы было чем подкрепиться во время игры, команда взяла с собой большую прямоугольную плитку шоколада размерами $w \times h$. У команды есть список из n пар чисел — размеры шоколадок, которые команда считает счастливыми. Прежде чем приступить к поеданию шоколадки, участники команды решили поделить имеющуюся плитку на счастливые шоколадки. Для этого они действуют следующим образом: сначала плитка шоколада ломается на 2 части по линии, строго параллельной одной из своих сторон, после чего каждую из полученных частей они могут продолжить ломать аналогичным образом.

Вам поручили определить, какое максимальное количество счастливых шоколадок команда сможет получить, действуя по данной схеме. Шоколадки, полученные поворотом счастливых, счастливыми не являются.

Формат входных данных

В первой строке входного файла заданы три целых числа w, h, n — размеры плитки шоколада и количество вариантов размера счастливых шоколадок соответственно ($1 \leq w, h \leq 300, 1 \leq n \leq w \times h$). В следующих n строках заданы пары целых чисел w_i, h_i — размеры счастливых шоколадок ($1 \leq w_i \leq w, 1 \leq h_i \leq h$).

Формат выходных данных

В единственную строку выходного файла выведите максимальное количество счастливых шоколадок, на которые можно разрезать данную плитку.

Примеры

chocolate.in	chocolate.out
21 11 4 10 4 6 2 7 5 15 10	15
9 12 5 1 12 2 6 3 4 4 3 6 2	9

Задача D. НВП

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

Числовая последовательность задана рекуррентной формулой: $a_{i+1} = (ka_i + b) \bmod m$.
Найдите её наибольшую возрастающую подпоследовательность.

Формат входных данных

Программа получает на вход пять целых чисел: длину последовательности n ($1 \leq n \leq 10^5$), начальный элемент последовательности a_1 , параметры k, b, m для вычисления последующих членов последовательности ($1 \leq m \leq 10^4$, $0 \leq k < m$, $0 \leq b < m$, $0 \leq a_1 < m$).

Формат выходных данных

Требуется вывести длину НВП.

Примеры

lis.in	lis.out
5 41 2 1 100	3

Замечание

В данном примере последовательность состоит из 5 элементов: $a_1 = 41$, $a_{i+1} = (2a_i + 1) \bmod 100$, то есть последовательность имеет вид 41, 83, 67, 35, 71.

Задача E. Кратчайший путь

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

Дан ориентированный взвешенный ациклический граф. Требуется найти в нем кратчайший путь из вершины s в вершину t .

Формат входных данных

Первая строка входного файла содержит четыре целых числа n, m, s и t — количество вершин, дуг графа, начальная и конечная вершина соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается тремя натуральными числами b_i, e_i и w_i — началом, концом и длиной дуги соответственно ($1 \leq b_i, e_i \leq n$, $|w_i| \leq 1000$).

Входной граф не содержит циклов и петель.

$1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из s в t . Если пути из s в t не существует, выведите «Unreachable».

Примеры

dag-shortpath.in	dag-shortpath.out
2 1 1 2 1 2 -10	-10
2 1 2 1 1 2 -10	Unreachable

Задача F. Игра

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

На листке записано в одну строку N ($2 \leq N \leq 100$) целых положительных чисел. Каждое число не превышает 200. Играют двое. За каждый ход можно зачеркивать крайнее число либо слева, либо справа. Зачеркнутое число добавляется к очкам игрока. N — четное. Игру начинает первый игрок. Необходимо вывести максимально возможную сумму очков для первого игрока при условии, что противник играет наилучшим образом.

Формат входных данных

В первой строке входного файла содержится одно целое число N ($2 \leq N \leq 100$). В следующих N строках записан исходный ряд чисел, по одному числу в строке.

Формат выходных данных

Выходной файл должен содержать единственное число – максимально возможную сумму очков для первого игрока при наилучшей игре второго игрока.

Примеры

game.in	game.out
4	16
4	
7	
2	
9	