

Задача А. 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 входного файла

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 выходного файла

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

Примеры

<code>chocolate.in</code>	<code>chocolate.out</code>
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 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 входного файла

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 выходного файла

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. 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 поддерев. 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 входного файла

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 выходного файла

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

Примеры

balance.in	balance.out
6 2 3 0 0 4 5 0 6 0 0 0 0	1
3 0 2 0 3 0 0	1

Каждая строка должна содержать два числа l_i и r_i , а также заглавную букву латинского алфавита, задающую цвет c_i и означает, что робот должен покрасить участок забора с l_i по r_i - доску в цвет c_i (если длина забора n , должно выполняться $1 \leq l_i \leq r_i \leq n$).

Примеры

paint.in	paint.out
5 ABBCA	3 1 5 A 4 4 C 2 3 B
2 AA	1 1 2 A

Задача D. Покраска забора

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

Мэр города Многоярославца решил построить перед своим домом забор из n деревянных досок и нанять лучшего маляра города для его покраски. Поскольку забор должен стать главной достопримечательностью города, лучший дизайнер города для каждой доски назначил тщательно выбранный цвет, в который она должна быть покрашена.

Для покраски главный маляр решил применить новейшую технологию, специально разработанную им для выполнения этого задания. Покраской забора будет заниматься специальный робот, который за один час может покрасить произвольный отрезок забора (набор соседних досок) в некоторый цвет. Поскольку задание должно быть выполнено как можно быстрее, требуется составить программу для робота, которая позволит достичь требуемой раскраски за минимальное время. Оставить какую-то из досок непокрашенной, естественно, запрещается.

Формат входного файла

В первой строке входного файла записано число n ($1 \leq n \leq 300$), где n количество досок в заборе. Вторая строка содержит строку из n символов, описывающую требуемую покраску забора. Цвета обозначаются заглавными латинскими буквами.

Формат выходного файла

В первой строке выходного файла выведите m - наименьшее возможное время покраски забора в часах. Следующие m строк должны содержать программу покраски для робота.