

## 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	

## Задача А. Деловые встречи

Имя входного файла: `meetings.in`  
Имя выходного файла: `meetings.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Алексей — успешный предприниматель, и в течение одного дня у него бывает много встреч с разными деловыми партнёрами. К сожалению, встречи бывают разные и не все приносят ему радость, после других же настроение улучшается. Также, на многие встречи не стоит приходить в слишком плохом или хорошем настроении — результат таких встреч может быть не таким, какой хочется Алексею.

К счастью, недавно Алексей научился оценивать своё настроение с помощью целых чисел. После этого для каждой встречи он оценил, при каком максимальном и минимальном настроении стоит на неё приходить, а также как изменится его настроение после этой встречи. Теперь он хочет распланировать порядок встреч так, чтобы в течение дня совершить максимальное число встреч.

Ваша задача — написать программу, которая по информации о всех встречах и настроении Алексея в начале дня находит порядок проведения встреч такой, что их количество при этом максимально.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $n$  и  $k$  ( $1 \leq n \leq 20$ ,  $-100 \leq k \leq 100$ ) — количество встреч и настроение Алексея в начале дня.

Следующие  $n$  строк содержат по три целых числа  $a_i$ ,  $b_i$  и  $c_i$  ( $-100 \leq a_i, b_i, c_i \leq 100$ ) — минимальное и максимальное настроение, при котором встреча возможна, и изменение настроения по окончании встречи, соответственно.

### Формат выходных данных

В первой строке выходного файла выведите число  $m$  — максимально возможно число встреч. В следующей строке выведите  $m$  целых чисел — номера встреч в порядке их проведения. Встречи пронумерованы в порядке описания во входном файле.

Если ответов с максимальным числом встреч несколько, выведите любой.

### Примеры

<code>meetings.in</code>	<code>meetings.out</code>
3 0 1 3 3 0 1 2 1 3 1	3 2 3 1
3 1 -10 -5 3 -5 5 -2 -3 2 1	2 2 3

## Задача В. Сеть

Имя входного файла: `network.in`  
Имя выходного файла: `network.out`  
Ограничение по времени: 1.3 секунды  
Ограничение по памяти: 256 мегабайт

В компьютерной сети вашей фирмы  $n$  компьютеров. В последнее время свитч, к которому они подключены, сильно барахлит, и потому не любые два компьютера могут связаться друг с другом. Кроме того, если компьютер  $a$  обменивается информацией с компьютером  $b$ , то никакие другие компьютеры не могут в это время обмениваться информацией ни с  $a$ , ни с  $b$ . Вам необходимо вычислить максимальное количество компьютеров, которые могут одновременно участвовать в процессе обмена информацией.

### Формат входных данных

В первой строке файла задано число  $n$  ( $1 \leq n \leq 18$ ). Далее идут  $n$  строк по  $n$  символов, причём  $j$ -й символ  $i$ -й строки равен 'Y', если  $i$ -й и  $j$ -й компьютеры могут обмениваться информацией, иначе он равен 'N'. Верно, что  $i$ -й символ  $i$ -й строки всегда равен 'N' и, кроме того, матрица символов симметрична.

### Формат выходных данных

Выведите максимальное количество компьютеров, которые могут одновременно участвовать в процессе обмена информацией.

### Пример

<code>network.in</code>	<code>network.out</code>
5 NYYYY YNNNN YNNNY YNNNY YNYYN	4

## Задача С. Леденящая игра

Имя входного файла: `game.in`  
Имя выходного файла: `game.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Чтобы попасть в команду к Шкиперу пингвин должен пройти ряд испытаний: полоса препятствий от Шкипера, спарринг с Рико, расшифровка кода от Прапора и задача от Ковальски.

Вы, пингвин-новобранец, успешно дошли до последнего испытания. Ковальски предлагает вам сыграть в следующую игру. Вам дается  $m$  наборов разноцветных льдинок, каждая одного из  $n$  цветов. Различные цвета обозначаются различными прописными буквами латинского алфавита. Вы можете взять какое-то подмножество этих наборов при условии,

что льдинка каждого цвета будет встречаться не более одного раза в этом подмножестве. Пусть вы выбрали  $k$  наборов с индексами  $i_1, i_2, \dots, i_k$ , тогда ваш выигрыш составляет  $\sum_{j=1}^k l_{i_j} - k$  баллов, где  $l_{i_j}$  — количество льдинок в наборе  $i_j$ .

Ковальски требует найти подмножество с максимальным количеством баллов.

От вас требуется найти любое подмножество, подходящее под условия Ковальски.

### Формат входных данных

В первой строке входного файла находится число  $n$  ( $1 \leq n \leq 17$ ) — количество различных цветов. Вторая строка входного файла содержит число  $m$  ( $1 \leq m \leq 200000$ ) — количество различных наборов льдинок. В следующих  $m$  строках перечислены сами наборы. Набор с номером  $i$  задаётся строкой из первых  $n$  строчных латинских букв. Длина каждой строки не больше 17 символов.

### Формат выходных данных

В первой строке выходного файла выведите  $k$  — количество наборов в ответе. Во второй строке выходного файла выведите  $k$  чисел — индексы наборов, входящих в ответ, в произвольном порядке.

### Примеры

game.in	game.out
1 3 aaa aaaa a	0
1 2 aaa aaaa	0
3 3 aba ab c	1 2

### Задача D. Симпатичные узоры 2

Имя входного файла: nice2.in  
Имя выходного файла: nice2.out  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Компания [BrokenTiles](#) планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр.

Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника  $n \times m$  метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата  $2 \times 2$  метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

### Формат входных данных

На первой строке входного файла находятся два натуральных числа  $n$  и  $m$ .  $1 \leq n \cdot m \leq 300$ .

### Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера  $n \times m$  по модулю  $2^{30} + 1$ . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

### Примеры

nice2.in	nice2.out
2 2	14
3 3	322

### Задача E. Логическое дерево

Имя входного файла: boolean.in  
Имя выходного файла: boolean.out  
Ограничение по времени: 0.5 секунда  
Ограничение по памяти: 256 мегабайт

Рассмотрим разновидность двоичного дерева, которую мы назовем логическим деревом. В этом дереве каждый уровень полностью заполнен, за исключением, возможно, последнего (самого глубокого) уровня. При этом все вершины последнего уровня находятся максимально слева. Дополнительно, каждая вершина дерева имеет ноль или двоих детей.

Каждая вершина дерева имеет связанное с ней логическое значение (1 или 0). Кроме этого, каждая внутренняя вершина имеет связанную с ней логическую операцию („И“ или „ИЛИ“). Значение вершины с операцией „И“ — это логическое „И“ значений её детей. Аналогично, значение вершины с операцией „ИЛИ“ — это логическое „ИЛИ“ значений её детей. Значения всех листьев задаются во входном файле, поэтому значения всех вершин дерева могут быть найдены.

Наибольший интерес для нас представляет корень дерева. Мы хотим, чтобы он имел заданное логическое значение  $v$ , которое может отличаться от текущего. К счастью, мы мо-

жем изменять логические операции некоторых внутренних вершин (заменить „И“ на „ИЛИ“ и наоборот).

Дано описание логического дерева и набор вершин, операции в которых могут быть изменены. Найдите наименьшее количество вершин, которые следует изменить, чтобы корень дерева принял заданное значение  $v$ . Если это невозможно, то выведите строку «IMPOSSIBLE» (без кавычек).

#### Формат входных данных

В первой строке входного файла находятся два числа  $n$  и  $v$  ( $1 \leq n \leq 10\,000, 0 \leq v \leq 1$ ) — количество вершин в дереве и требуемое значение в корне соответственно. Поскольку все вершины имеют ноль или двоих детей, то  $n$  нечётно. Следующие  $n$  строк описывают вершины дерева. Вершины нумеруются от 1 до  $n$ .

Первые  $(n - 1)/2$  строк описывают внутренние вершины. Каждая из них содержит два числа —  $g$  и  $c$ , которые принимают значение либо 0, либо 1. Если  $g = 1$ , то вершина представляет логическую операцию „И“, иначе она представляет логическую операцию „ИЛИ“. Если  $c = 1$ , то операция в вершине может быть изменена, иначе нет. Внутренняя вершина с номером  $i$  имеет детей  $2i$  и  $2i + 1$ .

Следующие  $(n + 1)/2$  строк описывают листья. Каждая строка содержит одно число 0 или 1 — значение листа.

#### Формат выходных данных

В выходной файл выведите ответ на задачу.

#### Примеры

boolean.in	boolean.out
9 1 1 0 1 1 1 1 0 0 1 0 1 0 1	1
5 0 1 1 0 0 1 1 0	IMPOSSIBLE