Задача A. Цвет глаз

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В далекой стране, населенной волшебными существами, генетики обнаружили уникальное явление: цвет глаз у детенышей напрямую зависит от цвета глаз их родителей. В этой стране существуют всего три цвета глаз: карий, зеленый и синий.

Генетики выяснили несколько правил, определяющих, какой цвет глаз может быть у ребенка:

- Если хотя бы у одного из родителей карие глаза, то у ребенка обязательно будут карие глаза.
- Иначе, если хотя бы у одного родителя зеленые глаза, то и у малыша будут зеленые.
- Если оба родителя имеют одинаковый цвет глаз, то и у ребенка будет такой же цвет.

У генетиков есть старинная книга со статистикой о цветах глаз родителей и детенышей. Но в ней могут встречаться некорректные записи или символ «`.`», который обозначает, что цвет глаз не был зафиксирован и может принимать любое из трех значений.

Ваша задача помочь генетикам проверить корректность записей о цветах глаз. Вам даны цвета глаз обоих родителей и их детеныша. Необходимо определить все возможные корректные сочетания цветов глаз родителей и детеныша в лексикографическом порядке. Если такие сочетания невозможны, выведите слово «Incorrect».

Формат входных данных

Вам даны разделенные через пробел три цвета глаз: двух родителей и их детеныша. Цвет глаз может принимать значения «`brown`» (карий), «`green`» (зеленый), «`blue`» (синий) или символ «`.`»

Формат выходных данных

Выведите все возможные корректные сочетания цветов глаз родителей и детеныша, разделенных через пробел, в лексикографическом порядке, каждая комбинация должна быть на отдельной строке. Или слово «`Incorrect`», если сочетание недостижимо.

Примеры

стандартный ввод	стандартный вывод
`blue blue blue`	`blue blue blue`
`blue brown .`	`blue brown brown`

`green . .`	`green blue green`
	`green brown brown`
	`green green green`

`blue blue brown`	`Incorrect`

Задача B. Большая мировая политика

Ограничение по времени: 1.5 секунд
Ограничение по памяти: 512 мегабайт

Вам дана карта мира, представляющая собой прямоугольник $n \times m$. Каждая клетка карты принадлежит одной из k стран. Множество клеток каждой страны связно по сторонам — это означает, что из любой клетки можно дойти до любой другой клетки той же страны, перемещаясь вниз, вверх, влево и вправо, не заходя на территорию другой страны.

В наступившие смутные времена у многих стран есть территориальные претензии друг к другу. Страна i считает своими историческими территориями все клетки, лежащие в минимальном прямоугольнике со сторонами, параллельными краям карты, содержащем все клетки, принадлежащие стране. Таким образом, у страны i есть территориальные претензии ко всем странам j, имеющим клетки внутри минимального прямоугольника страны i, кроме самой страны i.

Перед вами, как перед профессиональным аналитиком, поставлена задача: для каждой страны i выяснить, к скольким странам она имеет территориальные претензии.

Формат входных данных

Первая строка входных данных содержит три целых числа n, m и k ($1 \leqslant n, m \leqslant 2 \cdot 10^5$; $1 \leqslant k \leqslant nm \leqslant 2 \cdot 10^6$) — высоту карты, ширину карты и количество различных стран, соответственно.

Далее следуют n строк, i-я из них содержит m целых чисел $a_{i1}, a_{i2}, \ldots, a_{im}$ ($1 \leqslant a_{ij} \leqslant k$) — номера стран, которым принадлежат клетки $(i, 1), (i, 2), \ldots, (i, m)$ соответственно.

Гарантируется, что каждой из k стран принадлежит хотя бы одна клетка.

Гарантируется, что множество клеток каждой страны связно по сторонам.

Формат выходных данных

Выведите k целых чисел, i-е из которых равно количеству стран, к которым страна i имеет территориальные претензии.

Пример

стандартный ввод	стандартный вывод
3 4 4 1 3 3 2 1 2 2 2 1 1 1 4	2 1 0 0

Задача C. Головоломка

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Перед вами головоломка. Дана таблица из n строк и m столбцов, заполненная нулями и единицами. Разрешён только один тип хода: внутри любого столбца вы можете произвольно переставлять элементы, меняя порядок строк в этом столбце как угодно. Количество нулей и единиц в каждом столбце остаётся прежним.

Таких ходов разрешается сделать сколько угодно. Цель головоломки — получить как можно больше полностью одинаковых строк.

Определите максимальное число строк, которые можно сделать полностью одинаковыми с использованием описанных ходов.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leqslant n, m \leqslant 10^5$; $nm \leqslant 2 \cdot 10^5$) — количество строк и столбцов.

Каждая из следующих n строк содержит по m символов '0' или '1' — элементы исходной таблицы.

Формат выходных данных

Выведите одно целое число — максимальное количество одинаковых строк, которое можно получить.

Пример

стандартный ввод	стандартный вывод
3 4 0101 0010 0100	2

Задача D. Новые шарики

Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Это интерактивная задача.

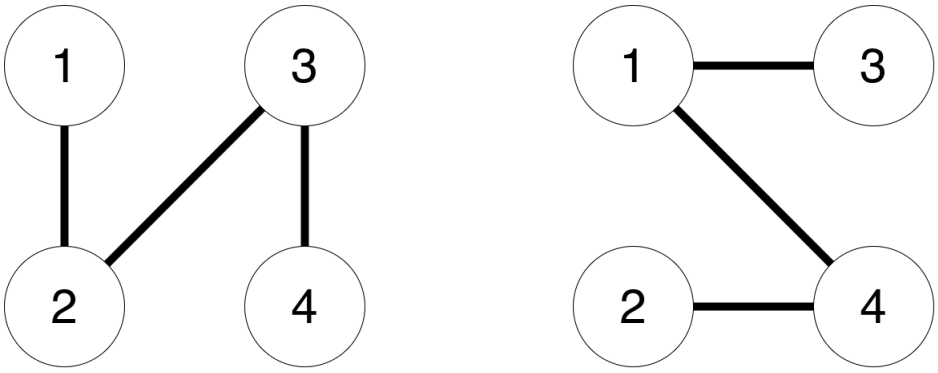
Обычно участникам ВКОШПа за каждую решенную задачу раздают шарики, но в этот раз за каждую решеную задачу команде будет выдаваться дерево! Напомним, что деревом называется связный неориентированный граф без циклов.

Так как в контесте n задач, то необходимо подготовить n различных деревьев, в каждом из которых будет по $2n$ вершин, пронумерованных от 1 до $2n$.

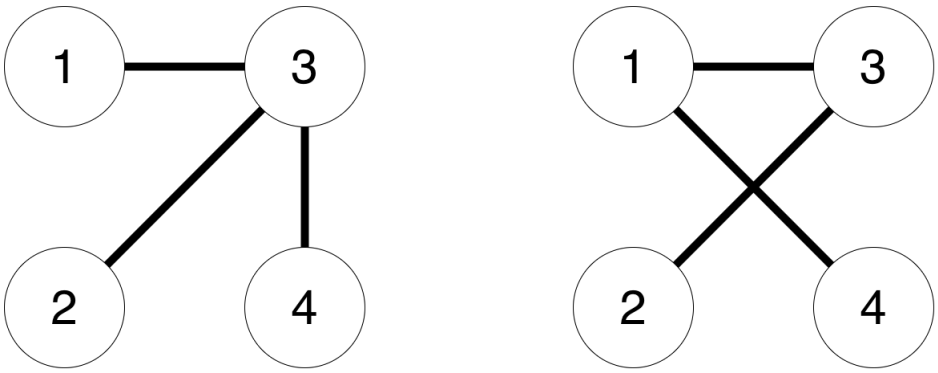
Однако оказалось, что деревья имеют тенденцию постоянно меняться. Если начать с некоторого дерева, то перед тем, как оно будет доставлено команде за решенную задачу, к дереву может не более $n-1$ раза быть применена следующая операция:

- Выбираются числа a, b, c, d так, что ребро (a, b) есть в дереве, а ребра (c, d) нет в дереве;
- После этого в из дерева удаляется ребро (a, b) и добавляется ребро (c, d). Гарантируется, что после данной операции граф останется деревом.

Вы заинтересованы в том, чтобы различать решенные задачи каждой команды, поэтому должны найти n деревьев так, чтобы их можно было отличить друг от друга даже после описанных модификаций.



Изображение выше показывают пример шариков из тестовых примеров для $n = 2$.



Деревья выше показывают пример задаваемых деревьев из тестовых примеров вашей программе.

Тестирование этой задачи организовано следующим образом. Сначала вы сообщаете программе жюри n деревьев, каждое из которых содержит $2n$ вершин.

После этого вам поочередно подается на вход q деревьев, каждое из которых получено из одного из ваших применением не более чем $n-1$ описанных выше операций. Для каждого дерева вы должны сказать, из какого оно было получено.

Протокол взаимодействия

Сначала ваша программа должна считать со стандартного ввода числа n и q ($1 \leqslant n, q \leqslant 100$).

Затем она должны вывести n деревьев, для каждого дерева необходимо вывести $2n - 1$ пар чисел — описание рёбер дерева.

После чего программа жюри q раз будет давать вам дерево, полученное операциями, описанными выше, и для каждого запроса вы должны будете вывести одно число x — номер исходного дерева, из которого получилось дерево жюри.

Пример

В примере сообщения разделены пустыми строками, чтобы лучше понимать, какое сообщение является ответом на какое. В реальном взаимодействии не следует выводить или ожидать на вводе пустые строки.

стандартный ввод	стандартный вывод
2 2	
	1 2 2 3 3 4 1 4 2 4 1 3
2 3 1 3 4 3	
	1
2 3 1 3 1 4	
	2

Замечание

После каждого действия вашей программы выводите перевод строки.

После каждого действия вашей программы делайте сброс потока вывода.

Если вы используете «`writeln`» в Паскале, «`cout << ... << endl`» в C++, «`System.out.println`» в Java, «`print`» в Python, «`Console.WriteLine`» в C#, то сброс потока вывода у вас происходит автоматически, дополнительно ничего делать не требуется. Если вы используете другой способ вывода, рекомендуется делать сброс потока вывода. Обратите внимание, что перевод строки надо выводить в любом случае.

Задача E. Опустошение массива

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

После сложного пути домой заяц Ксюша решила поиграть в игру на её зайкафоне.

Уровень в игре описывается массивом длиной n и целым числом x. За один ход в игре можно взять 2 соседних элемента массива, сумма которых равна x, и удалить их. После этого размер массива уменьшится на 2. Победа в уровне засчитывается в том случае, если после некоторого количества действий массив окажется пустым.

Очередной уровень показался ей очень сложным, и она даже засомневалась, возможно ли его пройти, — ведь может быть ситуация, что создатели игры не придумали больше уровней и сделали невозможным последний уровень. Ксюша обратилась к вам за помощью — подскажите ей, возможно ли пройти эту игру.

Формат входных данных

Первая строка содержит два целых числа n, x — размер массива и сумма удаляемых элементов ($1 \leqslant n \leqslant 3 \cdot 10^5$, $-10^9 \leqslant x \leqslant 10^9$).

Вторая строка каждого набора входных данных содержит n чисел a_i — элементы игрового массива ($-10^9 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Выведите «`Yes`», если Ксюша сможет пройти уровень, и «`No`» иначе.

Примеры

стандартный ввод	стандартный вывод
4 10 6 7 3 4	Yes
1 7 7	No
6 -3 -3 -9 6 3 -6 0	Yes
6 4 1 5 2 4 33 0	No

Задача F. Шоколадки Фибоначчи

Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Есть n шоколадок, i-я шоколадка имеет размер $w_i \times h_i$ и принадлежит либо Алисе, либо Бобу. В процессе игры шоколадки нельзя поворачивать на $90°$, то есть шоколадки $a \times b$ и $b \times a$ — разные шоколадки.

Выбирается непустое подмножество этих шоколадок, остальные шоколадки выбрасываются, а с выбранными шоколадками начинается игра между Алисой и Бобом. Игроки делают ходы по очереди, начиная с Алисы. В свой ход Алиса может сделать ровно одно из двух:

- съесть все кусочки одной из своих шоколадок (все части одной из своих шоколадок, то есть для какого-то i, что $o_i = 0$, все части этой шоколадки суммарного размера $w_i \times h_i$ (возможно, некоторые из этих частей получились в процессе деления другим игроком) больше не участвуют в игре).
- разделить один из текущих кусочков любой, не обязательно своей, шоколадки. Если разделяемый кусочек имеет размер $a \times b$, то в результате могут получиться два кусочка размера $x \times b$ и $(a - x) \times b$, где x — число Фибоначчи, такое что $0 < x < a$.

В свой ход Боб может сделать ровно одно из двух:

- съесть все кусочки одной из своих шоколадок (все части одной из своих шоколадок, то есть для какого-то i, что $o_i = 1$, все части этой шоколадки суммарного размера $w_i \times h_i$ (возможно, некоторые из этих частей получились в процессе деления другим игроком) больше не участвуют в игре).
- разделить один из текущих кусочков любой, не обязательно своей, шоколадки. Если разделяемый кусочек имеет размер $a \times b$, то в результате могут получиться два кусочка размера $a \times y$ и $a \times (b - y)$, где y — число Фибоначчи, такое что $0 < y < b$.

Игра заканчивается, когда у очередного игрока нет допустимых ходов. Игрок, который не может сделать ход, проигрывает.

Подмножества выбираются среди 2^n подмножеств; два подмножества считаются различными, если существует индекс $i \in \{1, \ldots, n\}$, такой, что шоколадка с номером i присутствует в одном подмножестве, но отсутствует в другом.

Ваша задача — вычислить количество непустых подмножеств шоколадок, для которых Алиса выигрывает при оптимальной игре обоих игроков. Необходимо найти остаток от деления этого количества на число 998 244 353.

Формат входных данных

Первая строка содержит одно целое число n — количество шоколадок ($1 \leqslant n \leqslant 100$).

Следующие n строк содержат по три целых числа: w_i, h_i и o_i — ширину и высоту i-й шоколадки, а также её владельца ($1 \leqslant w_i, h_i \leqslant 50$, $o_i \in \{0, 1\}$). Если $o_i = 0$, то i-я шоколадка принадлежит Алисе, а если $o_i = 1$, то она принадлежит Бобу.

Формат выходных данных

Выведите одно целое число: количество непустых подмножеств шоколадок, на которых Алиса выигрывает игру, взятое по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
2 1 1 0 1 1 1	1
4 2 2 0 1 2 1 2 1 1 1 1 0	6

Задача G. Магический ритуал

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В Академии Архимагов существует особая традиция: каждый выпускник должен в совершенстве овладеть ритуалом упорядочивания магических сфер.

На длинном алтаре размещены n сфер, занумерованных слева направо. Каждая сфера обладает заданной силой, и для успешного ритуала необходимо расставить их по неубыванию силы. Для этого можно менять местами сферы, за одно действие можно поменять местами две любые сферы.

Однако магический поток между двумя позициями i и j нестабилен, и для обмена местами сфер на этих позициях необходимо потратить $(i - j - 2)^2$ единиц маны.

Ритуал считается завершённым, когда сферы выстроены в правильном порядке. Нужно провести ритуал так, чтобы суммарный расход маны был минимальным.

Вам предстоит ответить на несколько независимых запросов о проведении ритуалов.

Формат входных данных

В первой строке дано целое число t ($1 \leqslant t \leqslant 2 \cdot 10^5$) — количество ритуалов.

Далее следуют описания ритуалов.

В первой строке каждого ритуала дано число n ($1 \leqslant n \leqslant 2 \cdot 10^5$) — число сфер.

Во второй строке заданы n целых чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^9$) — сила каждой сферы.

Гарантируется, что сумма n по всем ритуалам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого ритуала выведите одно целое число — минимальный расход маны для завершения ритуала.

Пример

стандартный ввод	стандартный вывод
3 3 3 2 1 3 2 1 2 4 4 3 2 1	0 1 2

Задача H. Плотина бобров

Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Это интерактивная задача.

Плотина бобров представляет собой прямоугольник, состоящий из $n \times m$ клеток, причем между некоторыми соседними клетками есть секретные проходы, по которым бобры умеют перемещаться. У плотины есть k выходов — клетки в прямоугольнике, в которых можно выплыть из плотины.

Ученые пытаются узнать устройство плотины бобров. Они уже выяснили ее структуру — точно известно, между какими соседними клетками есть проходы. Также известно, что граф проходов образует дерево, — между любыми двумя клетками существует единственный путь по проходам, посещающий каждую клетку не более одного раза. К сожалению, ученым пока почти ничего не удалось узнать про выходы — они знают только то, что $k \geqslant 1$, то есть из плотины существует хотя бы один выход.

Для того, чтобы найти выход из плотины, ученые решили использовать подопытных бобров. Бобры — умные животные, и они идеально знают устройство своей плотины, в том числе местонахождение всех выходов. Так как бобров для испытаний у ученых осталось немного, они могут провести эксперимент не более 48 раз.

Эксперимент заключается в следующем. Ученые берут двух бобров и выпускают их в клетки (i_1, j_1) и (i_2, j_2) прямоугольной плотины. После этого каждый из бобров поплывет по проходам к ближайшему выходу. Все происходит невероятно быстро, поэтому ученые успевают понять только одно — какой из бобров достиг выхода быстрее. В случае, если бобры достигли выхода за одно и то же время, результат эксперимента не определён, — первым может доплыть как первый бобер, так и второй.

Ученым необходимо изучить плотину бобров изнутри, поэтому им нужно обнаружить хотя бы один выход из плотины. Помогите им найти его.

Протокол взаимодействия

В начале ваша программа считывает параметры теста.

В первой строке содержатся два целых числа n, m — размеры плотины ($2 \leqslant n, m \leqslant 10^6$; $4 \leqslant n \cdot m \leqslant 10^6$).

В следующих n строках содержатся строки h_i длины $m-1$ из 0 и 1, описывающие проходы в плотине: если $h_{i,j} = 1$, то в плотине существует прямой проход между клетками (i, j) и $(i, j+1)$.

В следующих $n-1$ строках содержатся строки v_i длины m из 0 и 1, описывающие вертикальные проходы в плотине: если $v_{i,j} = 1$, то в плотине существует прямой проход между клетками (i, j) и $(i+1, j)$.

Затем ваша программа осуществляет взаимодействие с программой жюри, выполняя запросы и получая результаты проведения экспериментов. Вы можете выполнить действие, описанное в условии, не более 48 раз.

Чтобы задать вопрос, выведите строку в формате «`?` i_1 j_1 i_2 j_2», после чего в прямоугольную плотину будут запущены 2 бобра в заданных вами точках (i_1, j_1) и (i_2, j_2). Должны выполняться условия $1 \leqslant i_1, i_2 \leqslant n$; $1 \leqslant j_1, j_2 \leqslant m$.

В ответ программа жюри выводит:

- 1, если первый бобер доберется до выхода не позже второго;
- 2, если второй бобер доберется до выхода не позже первого;
- Любой из двух вердиктов выше, если бобры доберутся до выходов одновременно.

Чтобы вывести ответ на задачу, выведите строку «`!` i j», где в качестве i и j должны быть номера строки и столбца, в которых находится один из выходов. Этот вывод не учитывается в количестве запросов. Если выходов в плотине несколько, можно вывести любой из них. После этого интерактор выведет вердикт — «`Win`», если ваше предположение верно, и «`Lose`» в противном случае. Если ваш ответ был неверен, ваше решение получит вердикт `WA` (Wrong Answer) и завершится. Во

избежание получения некорректных вердиктов, считав информацию о том, что выведенный ответ неверен, ваше решение тоже должно завершиться.

Если в какой-то момент ваша программа превышает лимит в 48 запросов, ваша программа завершится с вердиктом `WA`.

После каждого действия вашей программы выводите перевод строки.

После каждого действия вашей программы делайте сброс потока вывода.

Если вы используете «`writeln`» в Паскале, «`cout << ... << endl`» в C++, «`System.out.println`» в Java, «`print`» в Python, «`Console.WriteLine`» в C#, то сброс потока вывода у вас происходит автоматически, дополнительно ничего делать не требуется. Если вы используете другой способ вывода, рекомендуется делать сброс потока вывода.

Обратите внимание, что в данной задаче интерактор может быть адаптивным, то есть состояние лабиринта всегда соответствует уже сделанным запросам, но в остальном может меняться в процессе работы.

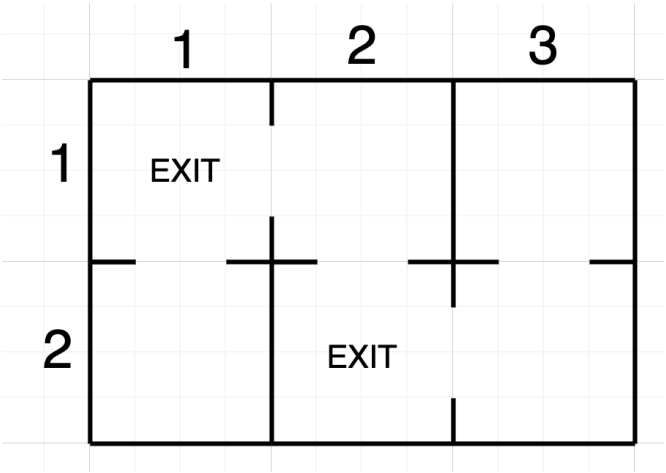
Пример

В примере сообщения разделены пустыми строками, чтобы лучше понимать, какое сообщение является ответом на какое. В реальном взаимодействии не следует выводить или ожидать на вводе пустые строки.

стандартный ввод	стандартный вывод
2 3 10 01 111	
	? 1 1 1 2
1	
	? 1 3 2 1
2	
	? 1 2 2 3
1	
	? 1 2 2 3
2	
	! 2 2
Win	

Замечание

В примере из условия лабиринт выглядит так:



Таким образом, выходы в плотине находятся в точках $(1, 1)$ и $(2, 2)$.

Задача I. Предсказывание позиции

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан отсортированный по возрастанию массив уникальных целых ключей x_i и некоторая целочисленная неотрицательная константа ε.

Необходимо разбить исходный массив ключей на минимальное количество подотрезков, чтобы на каждом подотрезке $[l \dots r]$ существовала некоторая линейная функция $f(x) = k \cdot x + b$, предсказывающая для ключа x_i его позицию на этом подотрезке — она равна $i - l$ — с ошибкой, не превышающей ε.

Формально, для j-го подотрезка должны существовать коэффициенты k_j и b_j (не обязательно целые), чтобы для всех ключей x_i на этом подотрезке $i \in [l_j \dots r_j]$ было верно:

$$|f_j(x_i) - (i - l_j)| \leqslant \varepsilon$$
$$f_j(x) = k_j \cdot x + b_j$$

Формат входных данных

В первой строке входных данных через пробел даны два целых числа: n, ε — количество ключей и максимально допустимая ошибка соответственно ($2 \leqslant n \leqslant 10^6$, $0 \leqslant \varepsilon \leqslant n$).

Во второй строке входных данных через пробел записаны уникальные ключи x_i в порядке возрастания ($-2 \cdot 10^9 \leqslant x_1 < x_2 < \dots < x_n \leqslant 2 \cdot 10^9$).

Формат выходных данных

В единственной строке выходных данных выведите одно целое число m — минимальное количество подотрезков, на которое вам удалось разбить исходный массив ключей так, чтобы удовлетворить требования.

Пример

стандартный ввод	стандартный вывод
8 0 1 2 3 4 7 10 13 16	2

Замечание

В примере можно разбить заданный массив ключей на два: $[1, 2, 3, 4]$ и $[7, 10, 13, 16]$. В первом позиция ключа предсказывается точно функцией $f(x) = x - 1$. Во втором позиция ключа предсказывается точно функцией $f(x) = (x - 7)/3 = 1/3 \cdot x - 7/3$.

Задача J. Странная сумма

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дано два неотрицательных целых числа n, x.
Также для всех $1 \leqslant l \leqslant r \leqslant n$ задано целое число $w_{l,r}$
Для массива целых чисел $A = [a_1, a_2, \ldots, a_n]$ определим

$$f(A) = \sum_{l=1}^{n} \sum_{r=l}^{n} w_{l,r} \cdot \min(a_l, a_{l+1}, \ldots, a_r).$$

Вам необходимо найти максимальное возможное $f(A)$ среди всех A, таких, что $a_1+a_2+\ldots+a_n = x$ и $a_i \geqslant 0$.

Формат входных данных

Входные данные в этой задаче содержат один или несколько тестовых примеров.

В первой строке записано одно целое число t — количество тестовых примеров ($1 \leqslant t \leqslant 50$).

Далее следуют описания t тестовых примеров.

В первой строке каждого тестового примера записаны два целых числа n и x ($1 \leqslant n \leqslant 50$, $1 \leqslant x \leqslant 10^9$).

В i-й из следующих n строк записаны $n - i + 1$ целых чисел $w_{i,i}, \ldots, w_{i,n}$ ($1 \leqslant w_{i,j} \leqslant 10^6$).

Формат выходных данных

Для каждого тестового примера выведите одно целое число: максимальное $f(A)$ среди всех A, таких, что $a_1 + a_2 + \ldots + a_n = x$ и $a_i \geqslant 0$.

Пример

стандартный ввод	стандартный вывод
3 5 10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 10 1 4 1000000000 1 2 3 4 5 6 7 8 9 10	30 10 14999999995

Задача K. Две косые дороги

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Читая условия очередной скучной задачи в каком-то командном контесте, который какие-то люди называют невероятно важным, вы предались ностальгии и мыслям о своём родном городе Берлодаре. Вспомнили, как вы с удовольствием гуляли вдоль его двух основных дорог — проспекта Холмса и улицы Ватсона, разбивающих город на четыре огромных квартала. Слегка усмехнулись, когда вы вспомнили, как после какого-то давнего урока в пятом классе, когда учитель математики рассказал вам про углы и расстояния, у вас загорелись глаза, и весь последовавший вечер вы вдоль и поперёк обошли главный перекрёсток города и выяснили, что ширина проспекта Холмса равна ℓ_H метров, ширина улицы Ватсона равна ℓ_W метров, а пересекаются они под углом в α градусов. Как вы дома раскопали карту города и с удовольствием на ней убедились, что ваши измерения были верны.

Скоро вы вернётесь в Берлодар на новогодние каникулы и захотите навестить своих четверых друзей. Они все живут в разных кварталах, так что вам придётся много раз переходить дорогу, а порой и обе, чтобы заглянуть ко всем друзьям. Планируя прогулку, вы задумаетесь: какие расстояния вам придётся преодолевать между этими четырьмя кварталами? Светофорный режим на этих двух улицах подразумевает, что зелёный свет для всех пешеходов (и, следовательно, красный свет для машин обеих дорог) загорается одновременно, так что, чтобы добраться между любой парой кварталов (даже находящихся напротив друг друга), вы пойдёте по кратчайшему отрезку, соединяющему их границы.

Есть $\frac{4\cdot(4-1)}{2} = 6$ пар кварталов. Напишите программу, которая выведет шесть чисел, равных попарным расстояниям между кварталами в Берлодаре.

Формат входных данных

В единственной строке входных данных находится три вещественных числа ℓ_H, ℓ_W и α — ширина проспекта Холмса в метрах, ширина улицы Ватсона в метрах и угол, под которым они пересекаются, в градусах ($0.01 \leqslant \ell_H, \ell_W \leqslant 1000$; $0.1 \leqslant \alpha \leqslant 90$). Каждое из этих трёх чисел будет подано на вход с точностью от нуля до трёх цифр после десятичной точки.

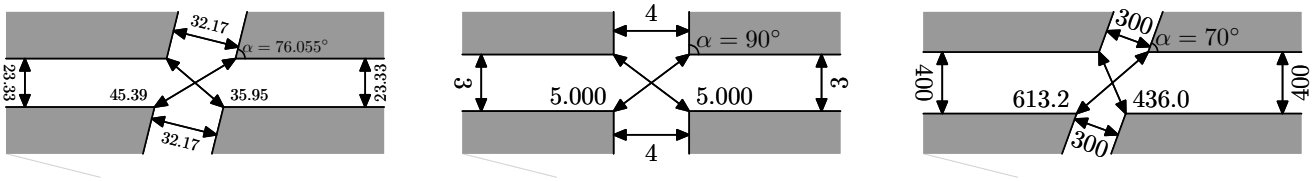
Формат выходных данных

Выведите в любом порядке шесть вещественных чисел — попарные расстояния между четырьмя кварталами Берлодара в метрах. Ваш ответ будет засчитан, если относительная или абсолютная погрешность не будет превышать 10^{-6}. Формально если $\{a_i\}_{i\in[1..6]}$ — ваш ответ, а $\{b_i\}_{i\in[1..6]}$ — ответ жюри, то он будет засчитан, если можно так переставить числа в массиве a, что после перестановки выполнено $\frac{|a_i-b_i|}{\max\{b_i,1\}} \leqslant 10^{-6}$ для каждого индекса $i \in [1..6]$.

Примеры

стандартный ввод	стандартный вывод
23.332 32.17 76.055	23.332 23.332000 32.17 32.170 35.95260 45.39539
3 4 90	5 4 3 3 4 5
400 300 70	400 300 400 300.0 613.251621126 436.073

Замечание



Задача L. Система уравнений с xor

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Алиса и Боб любят задачи на случайные числа. Однажды они придумали следующую задачу:

- Сначала Алиса выбирает случайное целое число x из диапазона от 1 до $2^{31} - 1$, все числа равновероятны.
- Затем Боб выбирает случайное целое число y из диапазона от 1 до $2^{31} - 1$, все числа равновероятны.
- Они считают произведение этих чисел $a = x \cdot y$ и их побитовый XOR $b = x \oplus y$.

Вам даны два полученных целых числа a и b. Найдите любую пару натуральных чисел x и y, таких, что:

$$xy = a \quad \text{и} \quad x \oplus y = b,$$

где $\oplus$ обозначает побитовую операцию XOR.

Напомним, что побитовое «`исключающее или`» ($\oplus$, xor) двух целых неотрицательных чисел устроено следующим образом: запишем оба числа в двоичной системе счисления, i-й двоичный разряд результата равен 1, если ровно у одного из аргументов он равен 1. Например, $(14 \text{ xor } 7) = (1110_2 \oplus 0111_2) = 1001_2 = 9$. Эта операция реализована во всех современных языках программирования, в языках C++, Java и Python она записывается как «`^`», в Паскале как «`xor`».

Формат входных данных

В первой строке входных данных содержится единственное целое число t ($1 \leqslant t \leqslant 200\,000$) — количество наборов входных данных.

В следующих t строках входных данных содержится по два целых числа a и b ($1 \leqslant a < 2^{62}$, $0 \leqslant b < 2^{31}$) — описание очередного набора входных данных.

Формат выходных данных

Для каждого набора входных данных выведите в отдельной строке два натуральных числа x и y, разделённых пробелом, таких, что $xy = a$ и $x \oplus y = b$.

Если существует несколько вариантов ответа, вы можете вывести любой из них.

Пример

стандартный ввод	стандартный вывод
2 21 4 9 0	7 3 3 3

Замечание

В этой задаче 100 тестов, учитывая пример из условия. Гарантируется, что во всех тестах, кроме примера из условия, $t = 200\,000$, а также числа a и b для каждого набора входных данных были получены с помощью выбора случайных чисел x и y Алисой и Бобом.

Задача M. Математический календарь

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Люся занимается в кружке по программированию по четвергам и в кружке по математике, занятия в котором проходят по понедельникам. На Новый год школьникам, которые особенно успешно занимаются на кружках, подарили красивый карманный календарь. Получила его в подарок и Люся.

Люся очень любит искать разные закономерности. Изучив календарь, она обнаружила, что в новом году число зимних четвергов на 2 больше, чем зимних понедельников. Люся поняла, что если знать про год это условие, а также то, что новый год не будет високосным, то по дню и месяцу можно однозначно определить, на какой день недели он приходится. Люся решила эту задачу и предлагает решить её вам.

Напомним, что зимними месяцами считаются декабрь, январь и февраль, в январе, марте, мае, июле, августе, октябре и декабре 31 день, в апреле, июне, сентябре и ноябре по 30 дней, в феврале невисокосного года 28 дней.

Формат входных данных

На единственной строке ввода дана строка — дата в виде дня и месяца (в формате `ДД-ММ`).

Формат выходных данных

Выведите целое число от 1 до 7 — день недели, на который придётся введённая дата.

Пример

стандартный ввод	стандартный вывод
16-12	3