Задача A. Опять физика

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Даня — начинающий физик. Сегодня он проводит масштабное исследование заряженных частиц.

Свое исследование Даня проводит на специальном полигоне, представляющем из себя плоскость с введенной на ней системой координат. Каждая точка плоскости исходно имеет заряд, равный 0.

Во время исследования Даня несколько раз совершает следующее действие: он запускает вдоль некоторой прямой на плоскости заряженную частицу. При этом к заряду каждой точки плоскости, лежащей на этой прямой, прибавляется величина заряда запущенной частицы.

Даня очень любопытный мальчик, и периодически у него возникает внезапный вопрос: каков суммарный заряд всех точек на некоторой выбранной им прямой? Ваша задача заключается в том, чтобы помочь Дане написать программу, которая бы отвечала на его вопросы.

Определим формально понятие суммы зарядов точек на прямой:

- Если на прямой все точки, кроме конечного числа, имеют положительный заряд, то их суммарный заряд равен $+\infty$.
- Если на прямой все точки, кроме конечного числа, имеют отрицательный заряд, то их суммарный заряд равен $-\infty$.
- Если на прямой конечное число точек с ненулевым зарядом, то их суммарный заряд равен сумме этих ненулевых зарядов.

Можно доказать, что в условиях данной задачи для любой прямой в любой момент времени будет иметь место ровно один из этих случаев.

Формат входных данных

Первая строка входных данных содержит единственное целое число n — количество запросов ($1 \leqslant n \leqslant 10^5$).

Каждая из следующих n строк содержит описание запроса Дани. Каждый запрос содержит целое число t — его тип ($t = 1$ или $t = 2$), а также три целых числа a, b, c — коэффициенты уравнения прямой ($-10^9 \leqslant a, b, c \leqslant 10^9$, $|a| + |b| > 0$).

Если $t = 1$, то запрос также содержит целое число q — заряд частицы, которую запускает Даня ($-10^9 \leqslant q \leqslant 10^9$). Этот тип запроса означает, что Даня запустил частицу заряда q вдоль прямой, задаваемой уравнением $ax + by + c = 0$.

Если $t = 2$, то Дане стало интересно, каков суммарный заряд всех частиц, лежащих на прямой, задаваемой уравнением $ax + by + c = 0$.

Формат выходных данных

Для каждого запроса 2 типа выведите суммарный заряд всех точек на выбранной прямой. Если он равен $+\infty$ или $-\infty$, выведите вместо числа единственную строку «`inf`».

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы и всех групп, от которых зависит данная группа.

Группа	Баллы	Дополнительные ограничения				Необх. группы
		n	|a|, |b|, |c|	|q|	Дополнительно	
1	20	$n \leqslant 100$	$\|a\|, \|b\|, \|c\| \leqslant 20$	$\|q\| \leqslant 100$	$a = 0$ или $b = 0$	—
2	15	$n \leqslant 1\,000$	$\|a\|, \|b\|, \|c\| \leqslant 50$	$\|q\| \leqslant 10^6$	$a = 1$ или $b = 1$	—
3	15	$n \leqslant 1\,000$	$\|a\|, \|b\|, \|c\| \leqslant 50$	$\|q\| \leqslant 10^6$	—	У, 1, 2
4	30	—	$\|a\|, \|b\|, \|c\| \leqslant 50$	—	—	У, 1, 2, 3
5	20	—	—	—	—	У, 1, 2, 3, 4

Примеры

стандартный ввод	стандартный вывод
2 1 0 1 0 239 2 1 0 0	239
5 1 1 0 1 1 1 0 1 1 1 2 1 0 2 2 0 1 1 2 0 2 1	1 inf 1
7 1 1 1 1 1 1 0 1 1 2 2 1 0 -1 1 1 2 1 3 2 1 1 1 2 1 1 -1 2 -1 2 2	3 inf 5 6

Задача B. Tax Collection

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Байтландия представляет собой прямоугольник $n \times m$, разделенный на $n \cdot m$ квадратных провинций. Недавно в Байтландии провели налоговую реформу, в результате которой для каждой провинции была зафиксировано число $a[i, j]$. Если $a[i, j] > 0$, значит провинция, находящаяся в квадрате (i, j) должна каждый месяц платить в бюджет $a[i, j]$ байткоинов. Если же $a[i, j] < 0$, то провинция (i, j) является дотационной и получает из бюджета $-a[i, j]$ байткоинов.

Для сбора налогов правительство разработало следующую схему. В одной из провинций будет построено здание казначейства. Каждый месяц из этого здания будет выезжать сборщик налогов. Он будет объезжать все провинции, собирая налоги и выдавая дотации, и возвращаться обратно. Его путь должен удовлетворять следующим свойствам:

- путь должен начинаться в провинции, в которой находится казначейство,
- путь должен заканчиваться в провинции, имеющей общую сторону с провинцией, в которой находится казначейство,
- каждая провинция должна быть посещена **ровно** один раз,
- соседние провинции в пути должны быть иметь общую сторону.

Правительство хочет выбрать провинцию для казначейства и путь для сборщика таким образом, чтобы для каждой дотационной провинции сборщик мог выдать им нужную сумму денег из уже собранных ранее. Помогите им построить такой путь или скажите, что это невозможно.

Формат входных данных

В первой строке заданы два целых числа n и m — размеры Байтландии ($2 \leqslant n, m \leqslant 300$).

Далее следует n строк, каждая из которых состоит из m целых чисел $a[i, j]$. Данные строки описывают провинции: $a[i, j]$ — значение, на которое изменится количество байткоинов у сборщика при посещении провинции, находящейся в позиции (i, j) ($1 \leqslant |a[i, j]| \leqslant 10\,000$).

Формат выходных данных

Если существует решение, то выведите $n \cdot m$ пар чисел — координаты провинций, которые должен посетить сборщик, в том порядке, в котором он должен их посетить. В случае, если решения не существует, выведите -1.

Система оценки

Тесты к этой задаче состоят из четырех групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы и всех групп, от которых зависит данная группа.

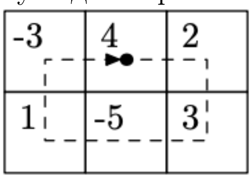
Подзадача	Баллы	Ограничения			Необх. подзадачи
		n	m	Доп. ограничения	
1	23	$n = 2$	$m \leqslant 300$	—	—
2	15	$n \leqslant 300$	$m \leqslant 300$	все $a_{i,j} > 0$	—
3	29	$n \leqslant 300$	$m \leqslant 300$	ровно одно $a_{i,j} < 0$	—
4	33	$n \leqslant 300$	$m \leqslant 300$	—	1, 2, 3

Примеры

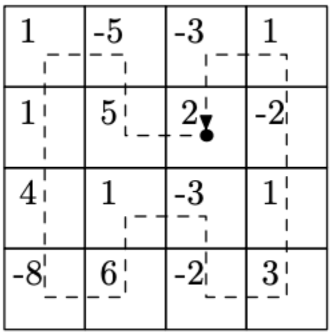
стандартный ввод	стандартный вывод
2 3 -3 4 2 1 -5 3	1 2 1 3 2 3 2 2 2 1 1 1
4 4 1 -5 -3 1 1 5 2 -2 4 1 -3 1 -8 6 -2 3	2 3 2 2 1 2 1 1 2 1 3 1 4 1 4 2 3 2 3 3 4 3 4 4 3 4 2 4 1 4 1 3
2 2 1 -2 -1 1	-1
3 3 1 1 1 1 1 1 1 1 1	-1

Замечание

Путь для первого примера:



Сумма, которая есть у сборщика после каждой провинции: 4, 6, 9, 4, 5, 2.
Путь для второго примера:



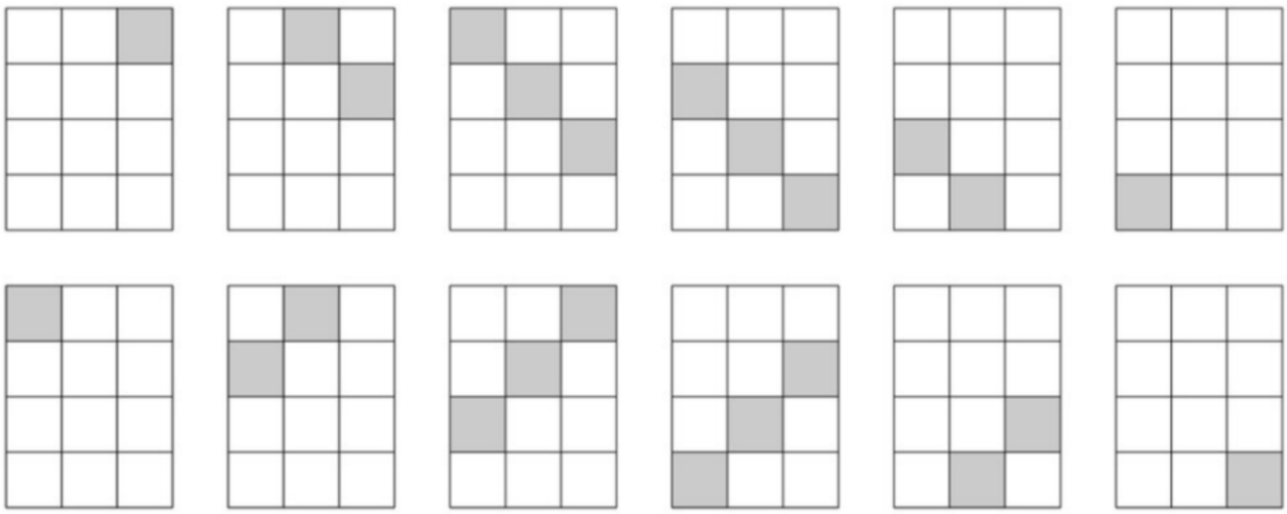
Сумма, которая есть у сборщика после каждой провинции: 2, 7, 2, 3, 4, 8, 0, 6, 7, 4, 2, 5, 6, 4, 5, 2.

Задача C. Покраска прямоугольника

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Сережа хочет покрасить прямоугольную таблицу, состоящую из m строка (пронумерованных от 0 до $m-1$) и n столбцов (пронумерованных от 0 до $n-1$). Изначально все клетки таблицы белые. На каждом шаге он выбирает какую-то диагональ и закрашивает все клетки на этой диагонали в свой любимый цвет. Однако стоимость покраски некоторых диагоналей может быть дороже других, вне зависимости от их длины. По заданной стоимости покраски каждой из диагоналей определите минимальную общую стоимость покраски всех клеток в таблице. Обратите внимание, что клетки можно перекрашивать дважды.

Прямоугольная сетка из m строк и n столбцов имеет $2m+2n-2$ диагоналей. Например, если $m=4$ и $n=3$, то всего существует 12 диагоналей:



Формат входных данных

В первой строке входного файла находится два целых числа m и n — размеры прямоугольника ($1 \leqslant m, n \leqslant 200\,000$).

Во второй строке находится $m+n-1$ целых чисел, которые задают стоимость покраски диагоналей по направлению $\searrow$, i-е число (для $1 \leqslant i \leqslant m+n-1$) относится к диагонали, в которой разность номера строки и номера столбца равна $i-n$. Таким образом, первое число относится к диагонали, состоящей только из одной клетки с координатами $(0, n-1)$ (строка 0, столбец $n-1$), второе число определяет стоимость покраски диагонали, включающей в себя клетки $(0, n-2)$ и $(1, n-1)$ и т. д.

В третьей строке находится $m+n-1$ целых чисел, которые задают стоимость покраски диагоналей по направлению $\nearrow$, i-е число (для $1 \leqslant i \leqslant m+n-1$) относится к диагонали, в которой сумма номера строки и номера столбца равна $i-1$. Таким образом, первое число относится к диагонали, состоящей только из одной клетки с координатами $(0,0)$ (строка 0, столбец 0), второе число определяет стоимость покраски диагонали, включающей в себя клетки $(0,1)$ и $(1,0)$ и т. д.

Все стоимости покраски это целые числа, лежащие в отрезке $[1, 10^9]$.

Формат выходных данных

Выведите минимальную стоимость покраски всей таблицы.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы и всех групп, от которых зависит данная группа.

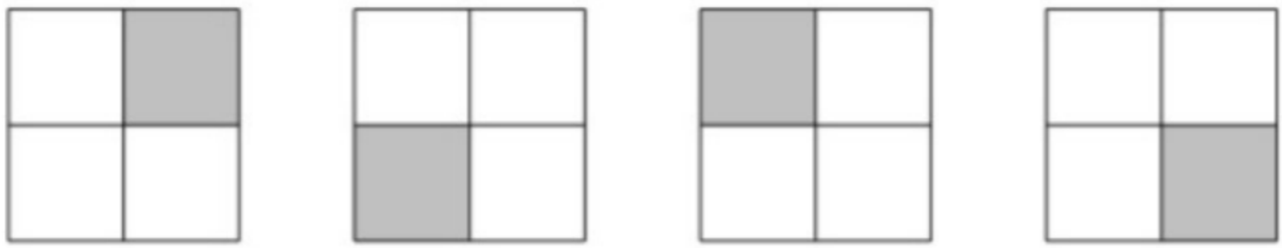
Группа	Баллы	Дополнительные ограничения		Необх. группы
		n, m	Дополнительно	
1	10	$m, n \leqslant 4$	—	У
2	10	$m, n \leqslant 10$	—	У, 1
3	10	$m, n \leqslant 20$	—	У, 1, 2
4	20	$m, n \leqslant 2\,000$	—	У, 1, 2, 3
5	10	—	$m = 1$	У
6	20	—	$m = n$	У
7	20	—	—	У, 1, 2, 3, 4, 5, 6

Примеры

стандартный ввод	стандартный вывод
2 2 1 3 1 1 3 1	4
4 3 2 3 9 3 4 3 2 3 3 1 2 4	14

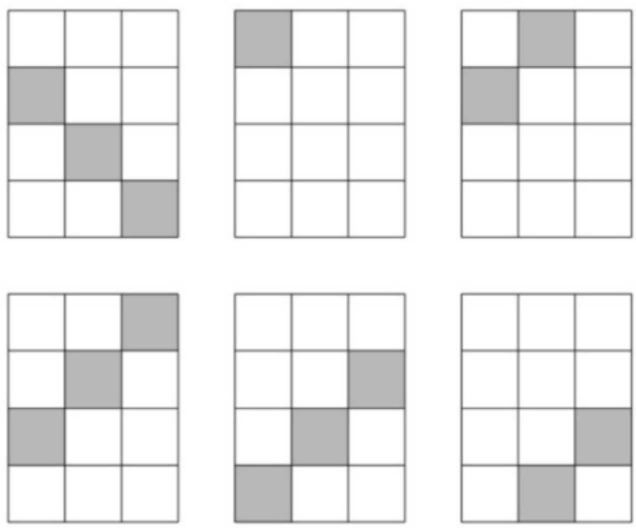
Замечание

В первом примере для минимизации стоимости должны быть покрашены следующие диагонали:



Покраска каждой из этих диагоналей стоит 1, соответственно итоговая стоимость равна 4.

Во втором примере минимальная стоимость получается покраской следующих диагоналей стоимостью $3, 2, 3, 3, 1, 2$, соответсвенно:



Задача D. Клубы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	512 мегабайт

Жители города X любят вступать в разные клубы. За последние годы число клубов в городе сильно увеличилось, и теперь там много клубов, в которых состоят одни и те же участники. Правительство города решило, что пора навести порядок с клубами. Было принято решение, что система клубов в городе должна удовлетворять следующему требованию:

(!) Для каждой пары жителей города должен быть хотя бы один клуб, такой что один из жителей в паре является членом этого клуба, а другой — нет.

Житель города может быть членом произвольного числа клубов, в частности может вообще не состоять в клубах.

Поскольку содержать клубы стоит денег, общее количество клубов должно быть минимально. Кроме того, члены каждого клуба собираются на общие встречи, а больших залов в городе нет. Поэтому после минимизации числа клубов, необходимо добиться того, чтобы количество членов в максимальном по количеству членов клубе должно быть минимально (в городе, разумеется, может быть несколько клубов с максимальным количеством членов).

В городе N жителей, пронумерованных от 1 до N.

Требуется написать программу, которая определит минимальное количество клубов, которое удовлетворяет условию (!). Также программа должна для каждого клуба назначить его членов, так, чтобы условие (!) выполнялось и максимальное количество членов в одном клубе было как можно меньше. Если возможных решений несколько, можно вывести любое из них.

Формат входных данных

Входные данные содержат одно число N — количество жителей в городе X ($2 \leqslant N \leqslant 100\,000$).

Формат выходных данных

На первой строке выведите два целых числа, разделенных пробелом — минимальное количество клубов и минимальное возможное количество членов в самом большом клубе.

Далее для каждого клуба следует вывести строку чисел, разделенных пробелами — первое число должно быть равно количеству членов клуба, а затем должны быть перечислены номера жителей, которые будут членами клуба. Номера жителей могут быть перечислены в любом порядке.

Система оценки

В этой задаче 20 тестов. Все тесты оцениваются независимо, каждый стоит 5 баллов.

Гарантируется, что в 10% тестов $N \leqslant 15$, в 20% тестов $N = 2^k$.

Пример

стандартный ввод	стандартный вывод
5	3 2 2 2 5 2 3 5 1 4

Замечание

Можно удовлетворить требование (!), используя не менее трех клубов, при этом в максимальном по размеру клубе два члена. Требование (!) можно также выполнить, например, таким распределением жителей по трем клубам: $2, 4, 5$, $3, 4$ и 5, но это не оптимальный ответ, поскольку в этом случае в максимальном клубе три члена.