

Задача А. Месть дракона

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Один сказочный король до смерти ненавидел драконов. Он собрал огромную армию и убил их всех до единого. Но он упустил из вида одно небольшое драконье яйцо, приняв его за камень, так что из этого яйца вылупился последний дракон. Быстро установив виновника геноцида остальных драконов, последний дракон сильно разозлился и разрушил всё королевство. Армии, чтобы от него отбиваться, совсем не осталось, поскольку она вся была потрачена на уничтожение всех остальных драконов.

Сказочное королевство представляет собой координатную плоскость. Король с остатками армии укрылся в каменном дворце, который находится в точке с координатами (x_p, y_p) . Дворец надёжно защищён, и дракон не способен его разрушить. Поняв это, дракон спрятался в точке с координатами (x_d, y_d) в надежде подкараулить короля, когда тот покинет дворец. Король вскоре узнал о местонахождении дракона и теперь пытается оценить опасность ситуации.

Если король будет находиться за пределами дворца, то в любой момент дракон может это заметить и вылететь за ним из своего логова. Вылет дракона немедленно будет замечен стражей, и король тут же начнёт равномерно прямолинейно двигаться в сторону дворца, пока не окажется в безопасности за его стенами. Дракон же жаждет мести, поэтому просто в каждый момент времени будет равномерно прямолинейно двигаться точно в направлении текущего местоположения короля. Король уже стар, а дракон в самом расцвете сил, так что скорость перемещения короля ровно в два раза меньше, чем скорость перемещения дракона. Король хотел бы определить, насколько сильно дракон ущемляет свободу его передвижения, поэтому желает узнать площадь безопасного участка плоскости, по которому он может разгуливать без опасений, что в случае вылета дракон догонит его раньше, чем он доберётся до дворца.

Формат входных данных

Первая строка содержит два целых числа через пробел: x_p и y_p ($-1000 \leq x_p, y_p \leq 1000$) — координаты дворца.

Вторая строка содержит два целых числа через пробел: x_d и y_d ($-1000 \leq x_d, y_d \leq 1000$) — координаты логова дракона.

Эти две точки не совпадают.

Формат выходных данных

Выведите единственное вещественное число — площадь безопасной части плоскости. Абсолютная или относительная погрешность не должна превышать 10^{-6} .

Пример

стандартный ввод	стандартный вывод
0 0	0.916297857297023
1 0	

Задача В. Адская мухобойка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У Жени дома летает очень много ос. Они постоянно летают под потолком в одних и тех же местах. Теперь Евгений отправился в магазин для покупки новой мухобойки. Все мухобойки имеют форму круга с различными радиусами. Женя — очень экономный студент, поэтому он решил купить самую дешёвую мухобойку — с минимально возможным радиусом, но Женя так же очень прагматичен, поэтому он купит только такую мухобойку, что с её помощью можно будет одним ударом убить всех ос. Помогите ему! Для простоты можете считать, что на потолке введена стандартная декартова система координат, и координаты ос постоянны. Помните, что ос у Жени действительно много.

Формат входных данных

В первой строке входного файла содержится число N — количество ос ($1 \leq N \leq 100\,000$). Далее содержатся координаты ос — пара целых чисел, не превосходящих по модулю 10^6 .

Формат выходных данных

В первой строке выходных данных выведите координаты точки, в которой Евгений должен нанести свой сокрушительный удар (это та точка, в которой будет расположен центр мухобойки). На следующей строке выведите одно число — минимальный радиус мухобойки, которого будет достаточно, чтобы уничтожить всех омерзительных ос. Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность не будет превышать 10^{-6} .

Пример

стандартный ввод	стандартный вывод
3	1.000 1.000
0 2	1.4142135623730951
0 0	
2 0	

Задача С. Целочисленные точки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Многоугольник (не обязательно выпуклый) на плоскости задан координатами своих вершин. Требуется подсчитать количество точек с целочисленными координатами, лежащих внутри него (но не на его границе).

Формат входных данных

В первой строке вводится n ($1 \leq n \leq 1000$) — число вершин многоугольника. В последующих n строках идут координаты (x_i, y_i) вершин многоугольника в порядке обхода по часовой стрелке. x_i и y_i — целые числа, по модулю не превосходящие 10^6 .

Формат выходных данных

Выведите одно число — количество искомым точек.

Пример

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

Задача D. Место встречи изменить нельзя

Имя входного файла: `rendezvous.in`
Имя выходного файла: `rendezvous.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

256 мегабайт

Даны N точек. Найдите такие две из них, что расстояние между ними минимально.

Формат входных данных

Первая строка входного файла содержит целое число N ($2 \leq N \leq 100\,000$) — количество точек. Каждая из следующих N строк содержит пару целых чисел X и Y , разделённых пробелом, — координаты ($-1\,000\,000\,000 \leq X, Y \leq 1\,000\,000\,000$). Все точки различны.

Формат выходных данных

Единственная строка выходного файла должна содержать координаты двух выбранных точек.

Пример

<code>rendezvous.in</code>	<code>rendezvous.out</code>
4	0 0
0 0	0 1
0 1	
1 1	
1 0	

Задача E. Диаметр точек

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На плоскости даны N точек. Вам требуется найти расстояние между двумя самыми удалёнными точками.

Формат входных данных

Первая строка содержит количество точек N , ($1 \leq N \leq 10^5$). Каждая из последующих N строк содержит два целых числа — координаты x_i и y_i . Координаты по модулю не превосходят 10^9 .

Формат выходных данных

Выведите в выходной файл расстояние между двумя наиболее удалёнными точками с максимально возможной точностью.

Примеры

стандартный ввод	стандартный вывод
5 0 0 2 2 1 1 0 2 2 0	2.828427124746
7 0 0 1 1 2 2 0 2 1 3 0 1 2 0	3.162277660168

Задача F. Не курить!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вася — хороший парень. Но у него есть плохая привычка — он курит. Все то время, сколько Петя дружит с Васей, он пытается отучить его от этого. Но ему это так и не удалось, потому что Вася не хочет бросать курить.

Недавно Петя придумал способ, как отучить своего друга от курения. Вася — неряха, поэтому его сигареты не лежат в пачке, а разбросаны по огромному столу. Петя хочет брать несколько сигарет в день незаметно для Васи. Вася не заметит пропажи сигарет, если в день будет пропадать не более одной сигареты. Кроме того, Петя должен брать только ту сигарету, которая пересекается с какой-нибудь другой сигаретой на столе. Помогите Пете узнать, сможет ли он начать реализацию своего плана.

Формат входных данных

Сигарета представляется как отрезок прямой. В первой строке входного файла записано число N ($1 \leq N \leq 125\,000$) — количество сигарет на Васином столе. Следующие N строк содержат описания сигарет: $(i + 1)$ -я строка содержит координаты концов i -й сигареты — целые числа x_1, y_1, x_2, y_2 ($-10\,000 \leq x_1, y_1, x_2, y_2 \leq 10\,000$).

Формат выходных данных

Если Петя не сможет взять ни одной сигареты, выведите в единственной строке выходного файла «NO», иначе выведите «YES»

Примеры

стандартный ввод	стандартный вывод
2 0 0 2 2 0 2 2 0	YES
4 0 0 3 1 2 0 3 -2 4 -1 7 2 -1 2 6 2	NO

Задача G. Простые внутренние точки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дан (совершенно необязательно выпуклый) простой N -угольник и K точек. Напомним, N -угольник называется простым, если не имеет ни самопересечений, ни самокасаний. Для каждой точки нужно определить, где она находится — внутри, на границе или снаружи.

Формат входных данных

В первой строке дано целое число T — количество тестов.
Далее идут T тестов. Тесты разделены переводом строки.
 N ($3 \leq N \leq 10^5$). Далее N точек — вершины многоугольника.
 K ($0 \leq K \leq 10^5$). Далее K точек — запросы.
Все координаты — целые числа, по модулю не превосходящие 10^9 .
Сумма $N \cdot K$ по всем тестам не превосходит $2 \cdot 10^6$.

Формат выходных данных

Для каждого запроса одна строка — `INSIDE`, `BORDER` или `OUTSIDE`.
Тесты следует разделять переводом строки.

Пример

стандартный ввод	стандартный вывод
1	INSIDE
4	BORDER
0 0	BORDER
2 0	OUTSIDE
2 2	
0 2	
4	
1 1	
0 0	
0 1	
0 3	

Задача Н. Внутренние точки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Дан (совершенно необязательно выпуклый) простой N -угольник и K точек. Напомним, N -угольник называется простым, если не имеет ни самопересечений, ни самокасаний. Для каждой точки нужно определить, где она находится — внутри, на границе или снаружи.

Формат входных данных

В первой строке дано целое число T — количество тестов.
Далее идут T тестов. Тесты разделены переводом строки.
 N ($3 \leq N \leq 10^5$). Далее N точек — вершины многоугольника.
 K ($0 \leq K \leq 10^5$). Далее K точек — запросы.
Все координаты — целые числа, по модулю не превосходящие 10^9 .
Суммарное количество N и K не превосходит $2 \cdot 10^6$.

Формат выходных данных

Для каждого запроса одна строка — INSIDE, BORDER или OUTSIDE.
Тесты следует разделять переводом строки.

Пример

стандартный ввод	стандартный вывод
1	INSIDE
4	BORDER
0 0	BORDER
2 0	OUTSIDE
2 2	
0 2	
4	
1 1	
0 0	
0 1	
0 3	

Задача I. Платные дороги

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Мэр одного большого города решил ввести плату за проезд по шоссе, проходящим в районе города, чтобы снизить объем транзитного транспорта. В районе города проходит n шоссе.

Но руководство области, в которой расположен город, воспротивилось планам мэра. Действительно — дальнобойщики представляют собой неплохой источник доходов для большого количества кафе и гостиниц в небольших городках.

В результате решили, что плата будет введена только на шоссе, которые проходят через город.

В городе используется развитая система метрополитена, всего в городе есть m станций метро. Решено было, что шоссе проходит через город, если либо одна из станций метро расположена непосредственно на шоссе, либо есть хотя бы одна станция с каждой стороны от шоссе.

Помогите теперь мэру определить, какие шоссе проходят через город.

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m — количество шоссе и количество станций метро, соответственно ($1 \leq n, m \leq 100\,000$).

Следующие n строк описывают шоссе. Каждое шоссе описывается тремя целыми числами a , b и c и представляет собой прямую на плоскости, задаваемую уравнением $ax + by + c = 0$ ($|a|, |b|, |c| \leq 10^9$).

Следующие m строк входного файла описывают станции метро. Каждая станция описывается двумя целыми числами x и y и представляет собой точку на плоскости с координатами (x, y) ($|x|, |y| \leq 10^9$).

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — количество шоссе, которые проходят через город. Вторая строка должна содержать номера этих шоссе в возрастающем порядке. Шоссе нумеруются от 1 до n в порядке, в котором они описаны во входном файле.

Примеры

стандартный ввод	стандартный вывод
4 2 0 1 0 1 0 1 1 1 0 1 1 -1 0 0 2 0	3 1 3 4
12 1 0 1 0 0 1 1 0 1 -1 1 0 1 1 0 -1 1 0 0 1 1 0 1 1 1 1 1 -1 1 -1 0 1 -1 1 1 -1 -1 0 0	4 1 6 7 10

Задача J. Извилистая ломаная

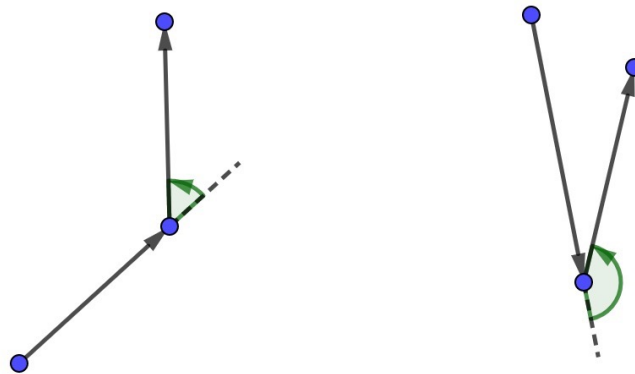
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У Васи есть n различных точек A_1, A_2, \dots, A_n на плоскости. Никакие три из них не лежат на одной прямой. Он хочет расположить их в некотором порядке $A_{p_1}, A_{p_2}, \dots, A_{p_n}$, где p_1, p_2, \dots, p_n — это некоторая перестановка чисел от 1 до n .

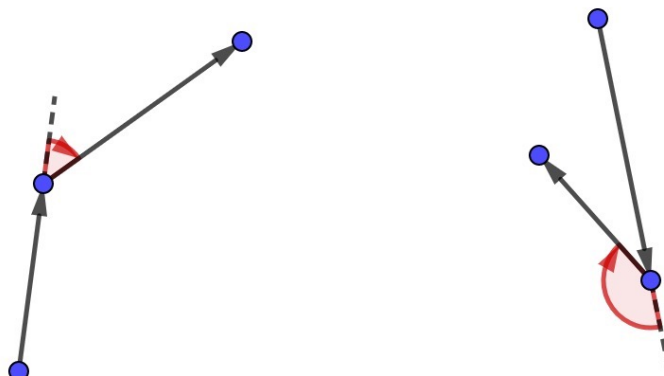
Сделав так, он нарисует ориентированную ломаную на этих вершинах, проведя направленные отрезки из каждой точки в следующую в выбранном порядке точку. То есть для всех $1 \leq i \leq n-1$ он проведет направленный отрезок из точки A_{p_i} в точку $A_{p_{i+1}}$. Он хочет, чтобы получившаяся ломаная удовлетворяла 2-м условиям:

- она будет несамопересекающейся, то есть любые 2 отрезка, которые не являются соседними, не имеют общих точек.
- она будет извилистой.

У Васи есть строка s , состоящая из $(n-2)$ -х символов “L” или “R”. Будем называть направленную ломаную извилистой, если её i -й поворот будет налево, если $s_i = \text{“L”}$ и направо, если $s_i = \text{“R”}$. Более формально: i -й поворот ломаной будет в точке $A_{p_{i+1}}$, в ней направленный отрезок из точки A_{p_i} в точку $A_{p_{i+1}}$ поменяется на направленный отрезок из точки $A_{p_{i+1}}$ в точку $A_{p_{i+2}}$. Обозначим вектор $\vec{v}_1 = \overrightarrow{A_{p_i}A_{p_{i+1}}}$ и вектор $\vec{v}_2 = \overrightarrow{A_{p_{i+1}}A_{p_{i+2}}}$. Тогда если для того, чтобы повернуть вектор \vec{v}_1 на наименьший возможный угол, чтобы его направление совпало с направлением вектора \vec{v}_2 надо сделать поворот против часовой стрелки, то будем говорить, что i -й поворот налево, а иначе направо. Для лучшего понимания посмотрите картинки, на которых изображены различные варианты поворотов:



На этой картинке изображены повороты налево



На этой картинке изображены повороты направо

Вам даны координаты точек A_1, A_2, \dots, A_n на плоскости и строка s . Найдите перестановку p_1, p_2, \dots, p_n чисел от 1 до n , такую что ломаная, которую нарисует Вася, будет удовлетворять двум заданным условиям.

Формат входных данных

В первой строке написано одно целое число n — количество точек ($3 \leq n \leq 2000$). В следующих n строках написаны по два целых числа x_i и y_i , разделённые пробелом — координаты точки A_i на плоскости ($-10^9 \leq x_i, y_i \leq 10^9$). В последней строке написана строка s из символов “L” и “R” длины $(n - 2)$. Гарантируется, что все точки различны и никакие три точки не лежат на одной прямой.

Формат выходных данных

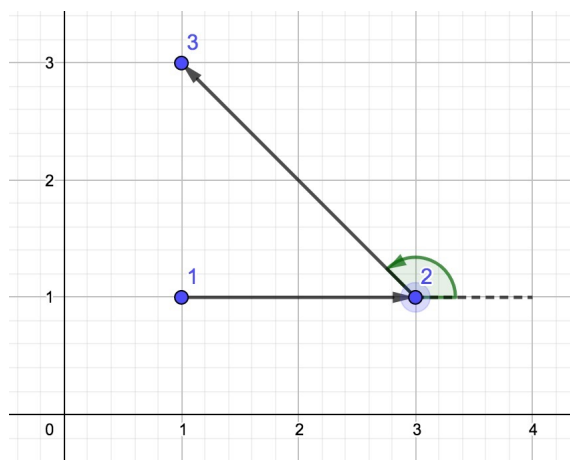
Если подходящей перестановки не существует выведите -1 . Иначе выведите n чисел p_1, p_2, \dots, p_n — найденную перестановку ($1 \leq p_i \leq n$ и все p_1, p_2, \dots, p_n различны). Если подходящих перестановок несколько, выведите любую.

Примеры

стандартный ввод	стандартный вывод
3 1 1 3 1 1 3 L	1 2 3
6 1 0 0 1 0 2 -1 0 -1 -1 2 1 RLLR	5 4 1 6 2 3

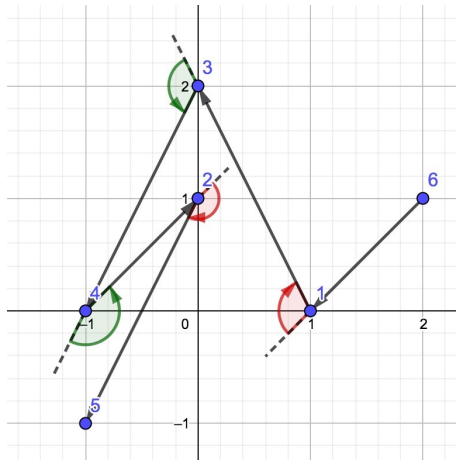
Замечание

Вот картинка, изображающая ломаную из 1 теста:



Как мы видим, эта ломаная несамопересекающаяся, а также извилистая, так как поворот в точке 2 налево.

Вот картинка, изображающая ломаную из 2 теста:



Задача К. Сыр

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Крешо купил вкуснейший сыр с перцем, но Степану перец не нравится, поэтому он хочет отрезать кусок, на котором не было бы перца. Сыр имеет форму выпуклого многоугольника, а каждая перчинка является точкой внутри него. Степан режет сыр только 1 раз. Он выбирает две вершины многоугольника, не являющиеся смежными, и режет по диагонали, соединяющей их. Затем Степан забирает ту из получившихся частей, на которой нет перца (ни внутри, ни на границе).

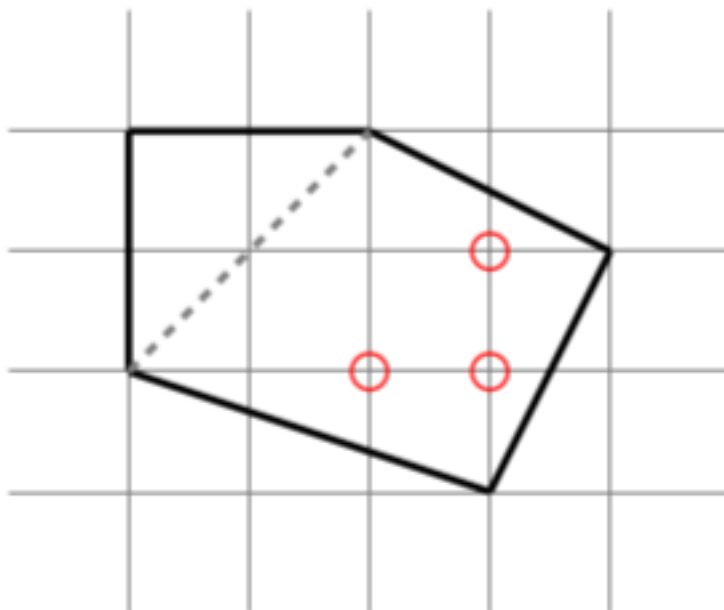


Рисунок соответствует первому тесту. Пунктирной линией показан разрез Степана.

Напишите программу, которая определит, может ли Степан отрезать кусок без перца. Если он может это сделать, выведите максимальную площадь куска, который может отрезать Степан.

Формат входных данных

Первая строка входного файла содержит одно целое число N ($3 \leq N \leq 3 \cdot 10^5$) – количество вершин в многоугольнике. Каждая из следующих N строк содержит два числа x_i и y_i – координаты i -й вершины. Следующая строка содержит одно число M ($1 \leq M \leq 3 \cdot 10^5$) – количество перчинок. Каждая из следующих M строк содержит два числа x_i и y_i – координаты i -й перчинки.

Вершины многоугольника заданы в порядке обхода против часовой стрелки и образуют выпуклый многоугольник. Никакие две подряд идущие стороны не параллельны.

Все перчинки расположены в различных точках и внутри многоугольника (они не расположены на стороне или снаружи многоугольника).

Все входные координаты по модулю не превосходят 10^9 .

Формат выходных данных

Выведите одно число — удвоенную максимальную площадь (это число всегда целое). Если отрезать кусок без перца невозможно, выведите 0.

Пример

стандартный ввод	стандартный вывод
5 0 1 3 0 4 2 2 3 0 3 3 2 1 3 1 3 2	4

Задача L. Полёт над озером

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

Петя любит делать снимки местности при помощи квадрокоптера. Однажды он отправился в путешествие к живописным берегам озера. Озеро может быть представлено в виде **не обязательно выпуклого** многоугольника на плоскости. Петя запустил квадрокоптер над озером на небольшом расстоянии от поверхности воды и сделал несколько снимков. Если квадрокоптер находится над точкой озера с координатами (x', y') , то на снимке будут видны все точки плоскости, y -координаты которых не превосходят y' . Для каждого снимка Петю интересует суммарная площадь частей озера, которые видны на этом снимке. Но он очень занят обработкой фотографий и просит Вас помочь с этой задачей.

Формат входных данных

В первой строке задано целое число n ($3 \leq n \leq 100\,000$) — количество вершин многоугольника, описывающего озеро.

Каждая из следующих n строк содержит по два целых числа x_i, y_i ($0 \leq x_i, y_i \leq 10^9$) — координаты вершин многоугольника в порядке обхода.

Гарантируется, что озеро представляет собой корректный невырожденный многоугольник, то есть не содержит самопересечений и самокасаний, также никакие три подряд идущие вершины многоугольника не лежат на одной прямой. Обратите внимание, что многоугольник может быть **невыпуклым**.

Следующая строка содержит целое число q ($1 \leq q \leq 100\,000$) — количество снимков.

Каждая из следующих q строк содержит по одному целому числу y'_i ($0 \leq y'_i \leq 10^9$) — y -координату точки озера, над которой находился квадрокоптер, чтобы сделать i -й снимок.

Формат выходных данных

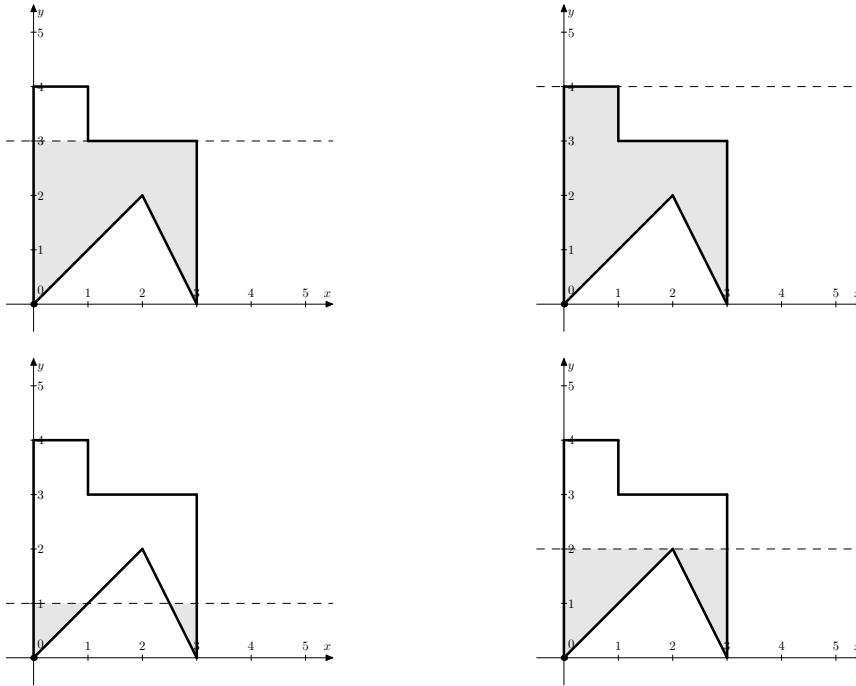
Выведите q чисел s_i — суммарную площадь озера, которую видно на i -й фотографии. Можно доказать, что ответ всегда представим в виде несократимой дроби $\frac{P}{Q}$. Вам нужно вывести $s_i \equiv P \cdot Q^{-1} \pmod{10^9 + 7}$. Иными словами, вам необходимо найти такое s_i ($0 \leq s_i < 10^9 + 7$), что $s_i \cdot Q \equiv P \pmod{10^9 + 7}$.

Примеры

стандартный ввод	стандартный вывод
7 3 0 3 3 1 3 1 4 0 4 0 0 2 2 4 1 3 2 4	750000006 6 3 7
11 5 7 8 7 9 5 12 7 14 7 14 5 12 5 12 2 1 2 1 9 3 9 10 5 8 6 3 13 7 2 1 9 4	33 500000062 45 11 61 55 0 0 61 22

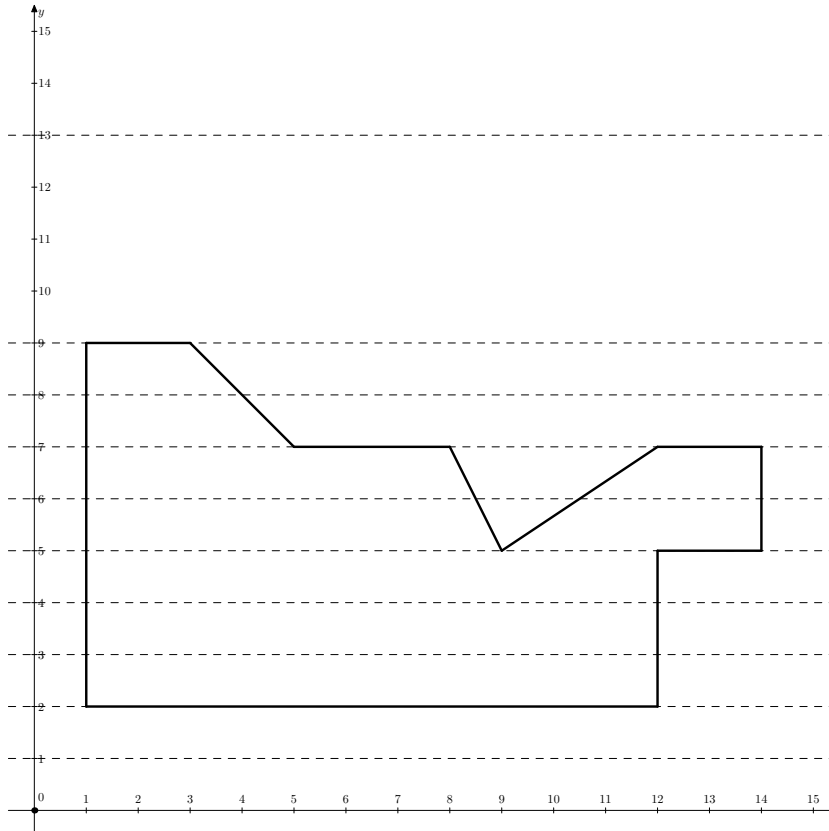
Замечание

В первом тесте из условия озеро и области снимков выглядят так:



На первой фотографии площадь видимой части озера равна $\frac{3}{4}$, на остальных снимках площадь видимой части озера является целым числом.

Во втором тесте из условия озеро и области снимков выглядят так:



На второй фотографии площадь видимой части озера равна $58\frac{1}{2} = \frac{117}{2}$, на остальных снимках площадь видимой части озера является целым числом.