Задача A. Натуральное деление

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

Вася очень любит натуральные числа, а особенно делить одно натуральное число на другое.

Вася утверждает, что взял два натуральных числа a и b ($0 < a < b < 10^6$), разделил a на b и получил непериодическую десятичную дробь с n знаками после запятой ($n \leqslant 17$). К сожалению, сами числа a и b он не помнит. Проверьте, могло ли такое быть, и если да, предъявите такие числа a и b.

Формат входных данных

Первая строка ввода содержит число n ($1 \leqslant n \leqslant 17$).

Вторая строка ввода содержит n цифр, которые шли в ответе после запятой. Гарантируется, что последняя цифра отлична от нуля.

Формат выходных данных

Если ответа не существует, выведите «NO».

Если ответ существует, выведите «YES», а затем два натуральных числа a и b, которые использовал Вася. Должно выполняться неравенство $0 < a < b < 10^6$. Если подходящих ответов несколько, можно вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
1 2	YES 1 5
2 69	YES 69 100
3 001	YES 1 1000

Задача B. Площадь треугольника

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

Чебурашка сказал Гене, что он нарисовал прямоугольный треугольник, и попросил помочь найти его площадь. Он назвал Гене длину гипотенузы и высоту, проведённую к ней.

Помогите Гене посчитать площадь такого треугольника либо скажите, что Чебурашка ошибся и такого треугольника не существует.

Формат входных данных

В первой строке дано одно целое число c $(1 \leqslant c \leqslant 10^4)$ — длина гипотенузы.

Во второй строке дано одно целое число h $(1 \leqslant h \leqslant 10^4)$ — длина высоты, проведённой к гипотенузе.

Формат выходных данных

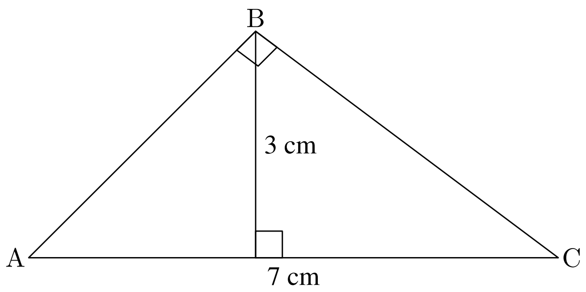
Выведите площадь треугольника. Если такого треугольника не существует, то выведите -1.

Примеры

стандартный ввод	стандартный вывод
7 3	10.5
10 6	-1

Замечание

Пример треугольника из первого примера.



Задача C. Число Чебурашки

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

Чебурашке подарили корзину, в которой было n апельсинов. Он решил поделиться ими с крокодилом Геной. Чтобы поделить апельсины, Чебурашка использует следующий алгоритм:

- Чебурашка кладет себе один апельсин и Гене один апельсин;
- Чебурашка кладет себе два апельсина, а Гене второй (то есть один);
- Чебурашка себе кладет три апельсина, а Гене третий (то есть один);
- И так далее..

Число n будем называть *числом Чебурашки*, если n апельсинов можно полностью поделить между друзьями указанным алгоритмом, а последний апельсин Чебурашка положил Гене.

Найдите k-е по возрастанию число Чебурашки.

Формат входных данных

На вход подается одно целое число k ($1 \leqslant k \leqslant 10^9$) — номер искомого числа Чебурашки.

Формат выходных данных

Выведите k-е число Чебурашки.

Примеры

стандартный ввод	стандартный вывод
1	2
2	5
3	9

Задача D. Ленивые разрезы

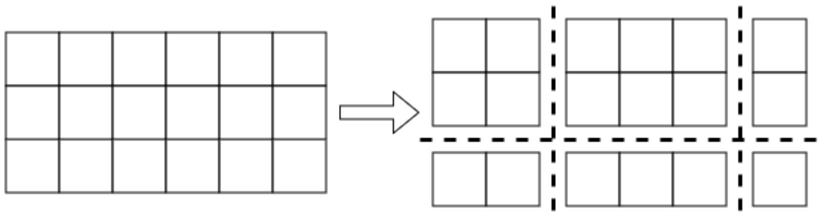
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Мальчику Антону нужно принести в школу на урок труда s клеточек. Антон очень ленивый, но очень любит уроки труда, поэтому сделать минимальное необходимое количество действий он готов.

Дома у Антона обнаружился клетчатый лист бумаги размера $n \times m$, при этом $s \leqslant n \cdot m$. Также у Антона есть лазерный резак, который работает следующим образом:

1. В резак устанавливается исходный лист бумаги
2. Резак делает x прямых вертикальных разрезов по границам клеток
3. Резак делает y прямых горизонтальных разрезов по границам клеток
4. Из резака вынимается $(x + 1) \cdot (y + 1)$ новых клетчатых листов бумаги

Например, из листа 3×6 за три разреза можно получить шесть листов с площадями 4, 2, 6, 2, 1 и 3.



Так как Антон ленивый, он хочет сделать минимальное возможное число разрезов, чтобы из полученных из резака листов бумаги можно было собрать набор с суммой площадей ровно s клеток, который он принесет в школу на урок труда.

Антон боится перетрудиться в процессе работы с резаком, поэтому просит вас помочь определить минимальное возможное число разрезов, которые ему придется сделать.

Формат входных данных

В единственной строке даны три целых числа n, m и s — размеры исходного клетчатого листа бумаги и необходимое для урока количество клеточек ($1 \leqslant n, m \leqslant 10^5$, $1 \leqslant s \leqslant n \cdot m$).

Формат выходных данных

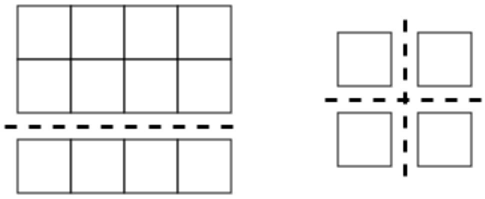
Выведите одно целое число — минимальное возможное число разрезов, которые нужно сделать, чтобы из полученных листов бумаги можно было собрать набор с суммой площадей ровно s клеток.

Примеры

стандартный ввод	стандартный вывод
3 4 8	1
2 2 3	2
10 9 90	0

Замечание

На рисунке ниже представлены примеры разрезов для первых двух тестов из примера.



В третьем тесте из примера $s = n \cdot m$, поэтому ничего резать не нужно.

Задача E. Декартов Кузнечик

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

Кузнечик по имени Декарт живет на Декартовой плоскости. Сегодня ему срочно нужно попасть из точки (x_0, y_0) в точку (x_1, y_1).

К сожалению, времени подумать у кузнечика нет, поэтому он строго решил прыгать n прыжков, при этом длина i-го прыжка должна быть в точности равна a_i, ведь ему как раз приснилась именно такая последовательность чисел.

Помогите кузнечику составить маршрут из точки (x_0, y_0) в точку (x_1, y_1) с помощью прыжков длины a_i, или подтвердите, что это невозможно.

Формат входных данных

В первой строке даны два целых числа x_0 и y_0 — координаты стартовой точки $(-1000 \leqslant x_0, y_0 \leqslant 1000)$.

Во второй строке даны два целых числа x_1 и y_1 — координаты конечной точки $(-1000 \leqslant x_1, y_1 \leqslant 1000)$. Обратите внимание, что стартовая и конечная точки маршрута кузнечика могут совпадать.

В третьей строке находится единственное целое число n — количество прыжков кузнечика $(1 \leqslant n \leqslant 10^5)$.

В четвертой строке заданы n целых чисел $a_1, a_2 \ldots a_n$ — желаемые длины прыжков $(1 \leqslant a_i \leqslant 1000)$.

Формат выходных данных

Если допрыгать из стартовой точки в конечную данной последовательностью прыжков невозможно, выведите в первой строке «`Impossible`».

Иначе, в первой строке выведите «`Possible`», а в следующих n строках выведите координаты точек. В i-й строке должны быть выведены два вещественных числа — координаты точки, в которой кузнечик окажется после i-го прыжка.

Если существует несколько решений, разрешается вывести любое. При этом следующие величины должны быть равны с относительной или абсолютной погрешностью не более 10^{-4}:

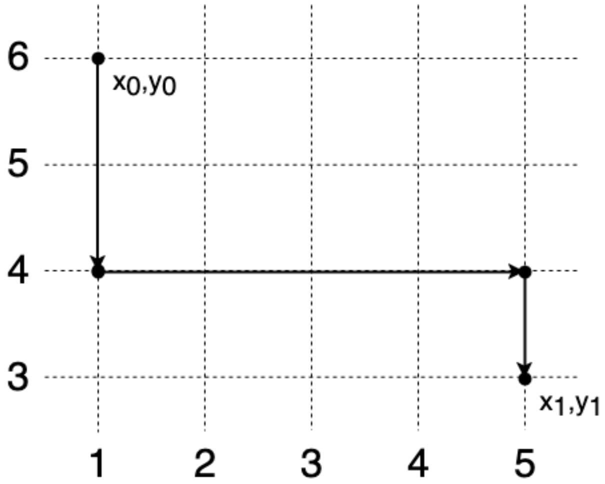
- Расстояние между $(i-1)$-й и i-й точками в ответе должно быть равно a_i;
- Расстояние между стартовой и первой в ответе точками должно быть равно a_1;
- Координаты n-й точки в ответе должны быть равны координатам конечной точки.

Примеры

стандартный ввод	стандартный вывод
1 6 5 3 3 2 4 1	Possible 1 4 5 4 5 3
0 0 6 8 5 2 3 4 5 6	Possible -1.20000000 -1.60000000 -3.00000000 -4.00000000 -0.60000000 -0.80000000 2.40000000 3.20000000 6.00000000 8.00000000
-10 10 10 -10 2 5 5	Impossible

Замечание

Один из возможных маршрутов кузнечика для первого теста из примера представлен на рисунке.



Задача F. Робот-доставщик

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

Прогрессивная бургерная «ITБургер» планирует запустить доставку своих бургеров с помощью автономных роботов-доставщиков, которые умеют самостоятельно планировать оптимальный маршрут доставки.

Для того, чтобы обучать робота, инженер Альбина построила испытательный полигон, который представляет собой координатную плоскость. В узлах координатной плоскости нанесено $n \times n$ световых меток в виде квадрата, соседние по столбцу и строке метки находятся на единичном расстоянии друг от друга. Левая нижняя метка находится в узле с координатами $(0, 0)$, а правая верхняя — в узле с координатами $(n - 1, n - 1)$.

Цель робота — построить маршрут, чтобы посетить все световые метки. Робот может начать движение в любой точке координатной плоскости с целыми координатами от -1000 до 1000. Он может двигаться только по прямой, поэтому его траекторией будет ломаная линия. Альбина считает, что робота можно запрограммировать так, чтобы он обошёл световые метки маршрутом в виде ломаной линии, содержащей ровно $2n - 2$ отрезков, причем начало и конец маршрута, а также все повороты будут в точках с целыми координатами от -1000 до 1000. Помогите роботу найти такой маршрут.

Формат входных данных

В первой строке ввода находится единственное целое число n ($3 \leqslant n \leqslant 100$) — количество световых меток в одной строке квадрата.

Формат выходных данных

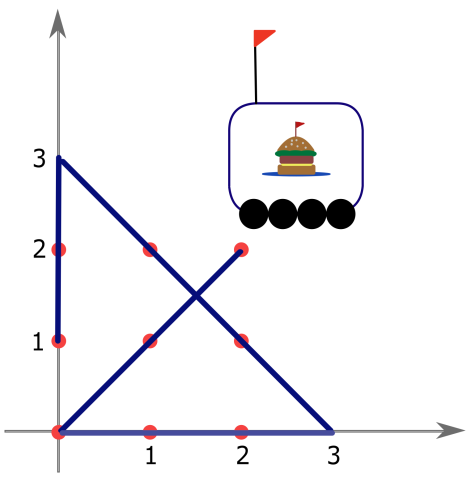
Выведите $2n - 1$ пар целых чисел — маршрут робота. Первая пара чисел — координата точки начала маршрута, следующие $2n-3$ пары чисел — точки, в которых робот делает поворот, последняя пара чисел — координаты конца маршрута.

Пример

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

Замечание

Маршрут робота при $n = 3$. Робот может начать в точке $(2, 2)$ и закончить в точке $(0, 1)$. Его траектория показана на рисунке.



Задача G. Поездка в Москву

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

Андрей живет в Санкт-Петербурге, однако ему часто приходится ездить в Москву по рабочим делам. Для перемещения между городами Андрей предпочитает использовать личный автомобиль и скоростную современную трассу, соединяющую два города. Так как у Андрея много работы, он не хочет тратить много времени на дорогу до Москвы. Поэтому он задумался о том, как минимизировать время поездки.

Расстояние от Москвы до Санкт-Петербурга составляет m единиц. Для удобства представим трассу, соединяющую города, как числовую прямую. Санкт-Петербург находится в точке 0, а Москва — в точке m. Также на трассе построены n заправочных станций, i-я из которых расположена на расстоянии x_i единиц от Санкт-Петербурга. Таким образом, i-я заправка будет расположена на числовой прямой в точке x_i.

Обозначим за t_i количество единиц времени, необходимых для того, чтобы заправиться на i-й станции. Для простоты будем считать, что время заправки не зависит от количества топлива, заправляемого в автомобиль.

Бак личного автомобиля Андрея вмещает в себя c литров топлива. Разумеется, расход топлива при езде по трассе зависит от скорости, с которой движется автомобиль. Андрей знает, что при езде со скоростью v единиц расстояния в единицу времени его автомобиль будет расходовать v литров топлива на одну единицу расстояния. Ограничения скорости на трассе нет, можно ехать с любой скоростью.

Изначально Андрей находится в Санкт-Петербурге, и его автомобиль заправлен до полного бака (то есть в баке автомобиля содержатся c литров топлива). Зная информацию о всех заправках на трассе, помогите Андрею вычислить минимальное количество единиц времени, которых хватит, чтобы добраться до Москвы. Обратите внимание, что Андрею не важно, сколько литров топлива останется в его автомобиле в момент, когда он окажется в Москве.

Формат входных данных

Первая строка содержит три целых числа n, m и c ($1 \leqslant n \leqslant 250\,000$, $n + 1 \leqslant m \leqslant 10^9$, $1 \leqslant c \leqslant 10^9$) — количество заправочных станций, расстояние от Санкт-Петербурга до Москвы, а также объем бака автомобиля Андрея.

Каждая из следующих n строк содержит два целых числа x_i и t_i ($0 < x_i < m$, $1 \leqslant t_i \leqslant 1\,000$) — расстояние от Санкт-Петербурга до i-й заправочной станции, а также время, необходимое для заправки автомобиля на ней.

Гарантируется, что $0 < x_1 < x_2 < \ldots < x_n < m$.

Формат выходных данных

Выведите одно вещественное число — минимальное количество единиц времени, необходимых для того, чтобы добраться до Москвы.

Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность не превосходит 10^{-6}.

Формально, пусть ваш ответ равен a, а ответ жюри равен b. Ваш ответ будет зачтен, если и только если $\frac{|a-b|}{\max(1,|b|)} \leqslant 10^{-6}$.

Пример

стандартный ввод	стандартный вывод
4 100 25 10 50 15 30 50 100 80 60	284.0000000000

Замечание

В первом примере оптимальный маршрут Андрея устроен следующим образом:

- Доехать от Санкт-Петербурга до второй заправочной станции, проехав 15 единиц расстояния. Данный участок можно проехать со скоростью $\frac{25}{15}$, чтобы израсходовать все топливо. Таким образом, путь займет 9 единиц времени;
- Заправить бак на второй станции, потратив 30 единиц времени;
- Доехать от второй заправочной станции до четвертой заправочной станции, проехав 65 единиц расстояния. Данный участок можно проехать со скоростью $\frac{25}{65}$, чтобы израсходовать все топливо. Таким образом, путь займет 169 единиц времени;
- Заправить бак на четвертой станции, потратив 60 единиц времени;
- Доехать от четвертой заправочной станции до Москвы, проехав 20 единиц расстояния. Данный участок можно проехать со скоростью $\frac{25}{20}$, чтобы израсходовать все топливо. Таким образом, путь займет 16 единиц времени.

Задача H. Разноцветный граф

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

Недавно Матвей на уроках информатики изучил графы. Граф представляет собой n вершин, некоторые из которых могут соединяться ребрами. Матвей — художник, и ему очень нравится рисовать. Когда он рисует граф, он изображает его ребра разными цветами. Будем обозначать цвета натуральными числами от 1 до 10^5.

Матвей придумал себе следующую задачу. Исходно в графе n вершин и нет ребер, а затем выполняется q запросов.

- $+\ v\ u\ c$ — добавить в граф ребро (v, u) цвета c. Гарантируется, что ребра цвета c между вершинами v и u не было.
- $-\ v\ u\ c$ — удалить в графе ребро (v, u) цвета c. Гарантируется, что ребро такого цвета было между вершинами u и v.

Так как Матвей — художник, он считает, что цвет c в графе является *превосходным*, если с каждой вершиной соединено не более одного ребра этого цвета. Определим *красоту* цвета c как количество рёбер такого цвета.

После каждого запроса ему стало интересно, какую суммарную красоту имеет граф по всем превосходным цветам.

Формат входных данных

В первой строке даны два целых числа n и q — количество вершин в графе и количество запросов, соответственно ($2 \leqslant n \leqslant 10^5$, $1 \leqslant q \leqslant 10^5$).

В последующих q строках даны описания запросов, по одному в строке ($1 \leqslant v, u \leqslant n$; $v \neq u$; $1 \leqslant c \leqslant 10^5$).

Формат выходных данных

После каждого запроса вам нужно вывести суммарную красоту графа по всем превосходным цветам.

Примеры

стандартный ввод	стандартный вывод
4 5 + 1 2 1 + 3 4 2 + 2 4 1 - 2 1 1 + 2 4 4	1 2 1 2 3
3 6 + 1 2 1 + 2 3 1 + 1 3 1 - 1 3 1 - 1 2 1 + 1 2 1	1 0 0 0 1 0

Замечание

В первом примере после первого запроса все цвета являются превосходными.

После второго запроса все цвета являются превосходными.

После третьего запроса цвет 1 не является превосходным, так как из вершины 2 исходит два ребра цвета 1. Цвет 2 является превосходным, и в графе есть одно ребро такого цвета.

После четвертого запроса все цвета являются превосходными.

После пятого запроса все цвета являются превосходными.

Задача I. «Галактический Таймлайн»

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

В Далёкой Галактике в Далёком будущем всё еще популярны старые настольные игры. Гуманоид Хёмуль играет в «Галактический Таймлайн».

Как известно, в игре «Таймлайн» есть карточки, на которых указан год и событие, которое произошло в этот год. Карточки необходимо выложить последовательно так, чтобы порядок событий получился правильным: каждое следующее событие не может быть раньше предыдущего.

В Далёкой Галактике экономят бумагу, поэтому в «Галактическом Таймлайне» карточки для игры двусторонние — на каждой карточке написано по два года и события — по одному на каждой из сторон карточек.

У Хёмуля есть n карточек, которые он выкладывает на стеклянный стол. Хёмуль заметил, что если выкладывать двусторонние карточки на стеклянный стол, то таймлайн формируется с двух сторон: сверху стола из лицевых сторон карточек и снизу стола из оборотных.

Изначально все карточки выложены в произвольном порядке в ряд слева направо. Хёмуль может выбрать любую карточку на столе и перевернуть ее. Также Хёмуль может выбрать две любые карточки на столе и поменять их местами, не переворачивая ни одну из них.

Хёмулю стало интересно, можно ли такими действиями выложить карточки так, чтобы события шли в правильном порядке слева направо как сверху стола, так и снизу стола. Помогите ему это сделать или определите, что выложить карточки в правильном порядке с обеих сторон невозможно.

Формат входных данных

В первой строке вводится число n ($1 \leqslant n \leqslant 10^5$). Во второй строке вводятся n чисел a_i ($1 \leqslant a_i \leqslant 10^9$) — годы, написанные на лицевой стороне i-й карточки. В третьей строке вводятся n чисел b_i ($1 \leqslant b_i \leqslant 10^9$) — годы, написанные на обратной стороне i-й карточки.

Формат выходных данных

Выведите информацию о карточках в том порядке, в котором их необходимо выложить.

В первой строке выведите n годов, находящихся на лицевой стороне карточек, во второй строке — n годов, находящихся на оборотной стороне карточек. Если выполнить цель игры невозможно, выведите -1.

Примеры

стандартный ввод	стандартный вывод
4 6 8 1 3 2 4 5 7	1 2 3 4 5 6 7 8
3 30 10 20 10 30 20	-1

Замечание

В первом примере Хёмуль может действовать, например, так. Сначала он переворачивает первую карту, затем переворачивает вторую карту. На столе оказываются выложены карты слева направо (лицевая и обратная сторона указаны через $|$): $2|6$, $4|8$, $1|5$ и $3|7$. Далее Хёмуль поменяет первую и третью карту, вторую и третью карту, третью и четвертую и получит правильный порядок с обеих сторон.

Задача J. Шахматный слон

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

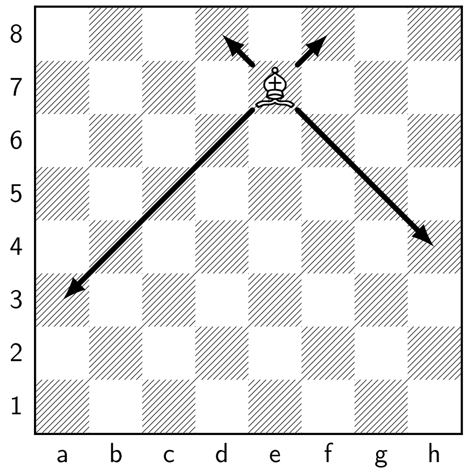
Это интерактивная задача.

Шахматная доска 8×8 занумерована стандартным способом: горизонтали — цифрами от 1 до 8, вертикали — строчными латинскими буквами от ‘a’ до ‘h’ так, что левый нижний угол — a1, а правый верхний — h8.

На шахматной доске находится шахматный слон, изначальная позиция которого неизвестна. Слон может перемещаться на любое число полей по диагонали, оставаясь в пределах доски. Ход слона задаётся следующим образом: сначала указывается смещение по горизонтали x (положительные значения — вправо), затем — смещение по вертикали y (положительные значения — вверх).

Если ход сделать удалось, слон перемещается на новое поле, и программа жюри выдаёт строку, состоящую из одного символа ‘+’. Если же поле назначения находится за пределами доски, слон остаётся на месте, и программа жюри выдаёт строку, состоящую из одного символа ‘-’.

Ваша задача — определить изначальную позицию слона, сделав не более 7-и попыток хода (удачных или неудачных).



На рисунке показана шахматная доска с шахматной нотацией. Слон расположен на поле e7. Направления возможных ходов показаны стрелками.

Протокол взаимодействия

Взаимодействие начинает программа участника выводом запроса. Каждый запрос имеет вид «? x y», где x и y — целые числа и $|x| = |y|$; $1 \leqslant |x|, |y| \leqslant 7$. В ответ на запрос программа жюри выдаёт строку, состоящую из символа ‘+’, если ход сделать удалось, и ‘-’, если ход сделать не удалось.

Когда программа участника готова вывести ответ, она вместо запроса должна вывести «! *pos*», где *pos* — изначальная позиция слона, заданная в стандартной шахматной нотации. После этого программа должна завершить своё исполнение.

Пример

стандартный ввод	стандартный вывод
`? 2 2`	
	`-`
`? 1 -1`	
	`+`
`? 2 2`	
	`+`
`? -1 1`	
	`-`
`? 1 1`	
	`-`
`? 1 -1`	
	`-`
`? -1 -1`	
	`+`
`! e7`	

Замечание

В примере ввода/вывода запросы программы участника и ответы программы жюри отформатированы пустыми строками, чтобы было видно, какой ответ соответствует какому запросу. В реальном взаимодействии программ участника и жюри пустых строк не будет.

Задача K. Магическое заклинание

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

Великий маг планирует составить магическое заклинание. Заклинание состоит из рун, которые будем обозначать целыми числами. Заклинание, которое планирует составить маг, представляет собой перестановку p, состоящую из n различных целых чисел от 1 до n.

Для составления заклинания маг может использовать специальный волшебный справочник, содержащий m страниц, пронумерованных от 1 до m, на каждой из которых записана одна руна, целое число от 1 до n. К сожалению, маг уже стар и не может сам переписывать руны из справочника; к счастью, у него есть k учеников, которых он может в некотором порядке призвать к себе и попросить переписать фрагменты справочника на один общий длинный свиток. Если призвать i-го ученика, то он по порядку перепишет на свиток руны со страниц справочника от l_i-й до r_i-й, включительно. Учеников можно призывать в любом порядке, но каждого ученика можно призвать не более одного раза.

Например, пусть в справочнике $m = 6$ рун записаны в следующем порядке: $[2, 1, 3, 1, 2, 4]$, у мага $k = 3$ ученика с $l_1 = 5$, $r_1 = 6$; $l_2 = 2$, $r_2 = 6$ и $l_3 = 1$, $r_3 = 3$.

Если призвать сначала третьего, а затем первого ученика, они выпишут на свиток руны в следующем порядке: $[2, 1, 3, 2, 4]$.

А если призвать сначала второго, затем первого и затем третьего ученика, то на свитке получится последовательность рун $[1, 3, 1, 2, 4, 2, 4, 2, 1, 3]$.

После того, как на свиток будет переписано некоторое количество рун, маг может вырезать любой непрерывный отрезок свитка. Он хочет призвать минимальное количество учеников, чтобы в итоге он мог, вырезав непрерывный отрезок получившегося свитка, получить искомое магическое заклинание.

В примере выше, если заклинание представляет собой перестановку $[1, 3, 2, 4]$, то его можно получить, призвав сначала третьего, а затем первого ученика, и вырезав из получившегося свитка $[2, 1, 3, 2, 4]$ руны со 2-й по 5-ю.

Помогите магу определить, какое минимальное количество учеников он должен призвать и в каком порядке их нужно призывать, чтобы составить искомое заклинание, если это возможно.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leqslant t \leqslant 10^5$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит три целых числа n, m и k ($1 \leqslant n \leqslant 3 \cdot 10^5$, $1 \leqslant m \leqslant 3 \cdot 10^5$, $1 \leqslant k \leqslant 3 \cdot 10^5$) — длина перестановки p, количество страниц в книге и количество учеников.

Следующая строка содержит n различных целых чисел p_i ($1 \leqslant p_i \leqslant n$) — перестановку p. Гарантируется, что все p_i попарно различны.

Следующая строка содержит m целых чисел a_i ($1 \leqslant a_i \leqslant n$) — руны, содержащиеся в книге.

Следующие k строк содержат по два целых числа, i-я из них содержит числа l_i и r_i ($1 \leqslant l_i \leqslant r_i \leqslant m$), задающие отрезок страниц, которые переписывает i-й ученик.

Гарантируется, что суммы n, m и k по всем наборам входных данных не превосходят $3 \cdot 10^5$ соответственно.

Формат выходных данных

Для каждого набора входных данных, если нельзя призвать учеников в некотором порядке так, чтобы перестановка p входила в выписанную на свиток последовательность рун как подотрезок, то выведите -1.

Иначе выведите число c — минимальное количество призываемых учеников.

В следующей строке выведите c чисел — номера учеников в порядке, в котором их следует призывать. Ученики нумеруются целыми числами от 1 до k в порядке перечисления во входных данных. Все номера учеников должны быть попарно различными.

Если существует несколько оптимальных ответов, выведите любой из них.

Пример

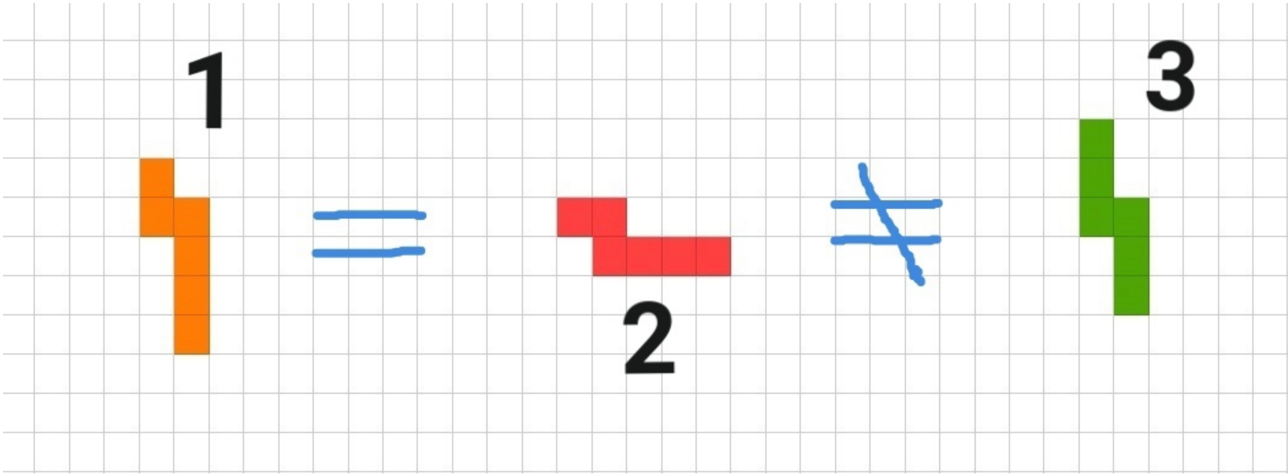
стандартный ввод	стандартный вывод
2 4 6 3 1 3 2 4 2 1 3 1 2 4 5 6 2 6 1 3 3 4 1 3 1 2 2 1 3 1 1 4	2 3 1 -1

Задача L. Мультимино

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

Назовём n-миношкой связную фигуру, составленную из n квадратиков 1×1. Скажем, что две n-миношки совпадают, если одну из них можно перевести в другую, используя операции сдвига, поворота и отражения.

Например, на рисунке снизу красная гексаминошка (номер 2) совпадает с оранжевой (номер 1), и они обе не равны зелёной (номер 3).



По данному вам числу n разрежьте клетчатое поле размера $n \times n$ квадратиков на n попарно различных непересекающихся n-миношек или сообщите, что такого разрезания не существует.

Формат входных данных

В единственной строке входных данных находится целое число n ($1 \leqslant n \leqslant 100$).

Формат выходных данных

Выведите единственное число -1, если нужного разбиения не существует.

Иначе пронумеруем n-миношки натуральными числами от 1 до n. Выведите n строк по n чисел в каждой, соответствующие разбиению поля на n-миношки по принципу, что клетка в i-ом столбце и j-ой строке принадлежит n-миношке номер a_{ij} ($1 \leqslant a_{ij} \leqslant n$).

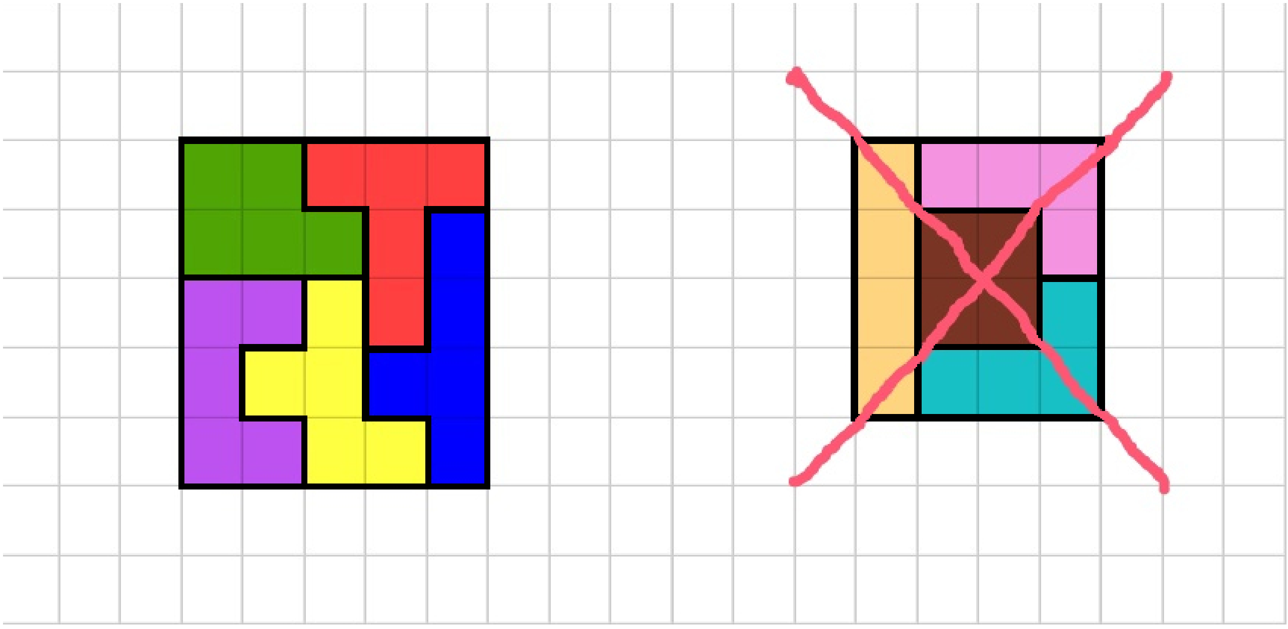
Все n-миношки в разбиении должны быть связны и попарно различны между собой. Если для поля существует несколько корректных разбиений, выведите любое из них.

Примеры

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

Замечание

В первом примере разбиение выглядит как на рисунке слева.



Во втором примере рисунок справа не будет корректным разбиением, так как сиреневая (верхняя) и бирюзовая (нижняя) n-миношки могут быть переведены друг в друга сдвигом, поворотом и симметрией, а следовательно, равны.

Задача M. Три точки

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

Чебурашка, крокодил Гена и старуха Шапокляк решили переехать в пустыню. Они нашли три одинаковых дома, которые располагаются в координатах x_1, y_1, x_2, y_2, x_3 и y_3, соответственно. Крыса Лариска получила задание — рассчитать координаты и прорыть для проведения водопровода прямую траншею бесконечной длины таким образом, чтобы минимизировать суммарное расстояние от домиков до этой траншеи.

Помогите ей определить координаты двух точек, которые лежат на этой прямой. Траншея может проходить под одним или несколькими домами.

Формат входных данных

В единственной строке даны шесть целых чисел x_1, y_1, x_2, y_2, x_3 и y_3 — координаты трёх домов, соответственно ($-10^9 \leqslant x_1, y_1, x_2, y_2, x_3, y_3 \leqslant 10^9$). Гарантируется, что дома находятся в различных точках.

Формат выходных данных

Выведите целые координаты двух точек, через которые будет проходить траншея. Точки не должны совпадать. Координаты не должны превышать 10^9 по модулю.

Пример

стандартный ввод	стандартный вывод
0 0 10 10 5 6	1 1 3 3