

Задача А. Телесъёмка

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

Для съёмки финального матча «Ротор–Закат» была нанята лучшая съёмочная группа. Несмотря на то, что их работа была выполнена на высочайшем уровне, после просмотра записи выяснилось, что один из игроков Заката ни разу не попал в кадр. Но тренеру интересны действия всех игроков, в том числе и того, которого не оказалось на записи.

Для упрощения задачи будем считать, что игровое поле представляет собой клетчатый прямоугольник $w \times h$ клеток. Каждую секунду игрок обязательно перебегает в соседнюю по стороне клетку. Съёмка же является последовательностью из n кадров, в каждом из которых видно какой-то подпрямоугольник поля с противоположными углами в (x_{i_1}, y_{i_1}) и (x_{i_2}, y_{i_2}) . Так как известно, что этот игрок ни разу не попал в кадр, содержимое кадров не важно, важно лишь то, какой участок поля снимался в каждый момент времени.

Ваша задача — помочь тренеру и восстановить какой-либо маршрут, по которому мог перемещаться игрок.

Формат входного файла

В первой строке заданы натуральные числа w и h ($1 \leq w, h \leq 300$) — ширина и длина поля. Во второй строке задано число n ($1 \leq n \leq 300$) — число кадров съёмки.

В следующих n строках задано по четыре числа $x_{i_1}, y_{i_1}, x_{i_2}$ и y_{i_2} ($1 \leq x_{i_1} \leq x_{i_2} \leq w, 1 \leq y_{i_1} \leq y_{i_2} \leq h$) — прямоугольник, соответствующий видимой на i -м кадре части поля.

Формат выходного файла

Выведите n строк, в каждой из которых должно быть по два числа x_i и y_i ($1 \leq x_i \leq w, 1 \leq y_i \leq h$) — координаты клетки, в которой мог оказаться этот игрок на i -й секунде. Выведите «Impossible», если не могло быть такой ситуации, что игрок не попал ни на один из кадров.

Примеры

<code>broadcast.in</code>	<code>broadcast.out</code>
3 2	1 2
5	2 2
1 1 2 1	2 1
2 1 2 1	3 1
2 2 3 2	3 2
2 2 3 2	
2 1 2 2	

Задача В. Берляндский футбольный союз

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

В третьем дивизионе чемпионата Берляндии по футболу играют n команд из n разных городов. Известно, что команды на игры ездят по *специальным* дорогам, построенным Берляндским футбольным союзом. Каждая дорога соединяет два разных города и по каждой дороге можно двигаться в обоих направлениях. У каждой дороги есть своя длина, измеряемая в километрах.

Система дорог устроена так, что между любыми двумя городами существует ровно один путь по специальным дорогам, проходящий через каждый город не более одного раза. Футбольные чиновники и эксперты также передвигаются только по этим дорогам.

Берляндский футбольный союз хочет построить штаб-квартиру в одном из этих n городов. Известно, что каждый месяц из штаб-квартиры футбольного союза будут выезжать n экспертов: в каждый город по одному эксперту, которые будут проверять качество полей и стадионов. Эксперту, который поедет в город, который находится в d километрах от штаба футбольного союза, будет оплачиваться премия в размере d^2 берляндских рублей.

Глава футбольного союза — очень экономный человек. Поэтому он хочет построить штаб-квартиру в таком городе, чтобы суммарная премия экспертам, выезжающим из него, была минимальна. Он попросил вас предоставить ему список городов-кандидатов, в которых можно будет построить штаб-квартиру Берляндского футбольного союза.

Формат входного файла

В первой строке входного файла задано целое число n ($1 \leq n \leq 10^5$) — количество городов в третьем дивизионе чемпионата Берляндии по футболу. В следующих $n - 1$ строках заданы специальные дороги, построенные футбольным союзом. Каждая дорога задается тремя целыми числами: u, v и w ($1 \leq u, v \leq n, 1 \leq w \leq 100, u \neq v$) — города u и v , соединенные этой дорогой, и ее длина w в километрах.

Формат выходного файла

В первую строку выходного файла выведите целое число k — количество городов-кандидатов, которые удовлетворяют условию главы футбольного союза Берляндии. Во второй строке выведите k чисел через пробел: номера кандидатов в любом порядке.

Примеры

<code>bfs.in</code>	<code>bfs.out</code>
6	1
6 3 35	6
5 2 12	
4 5 31	
4 6 14	
3 1 40	

Задача С. Автомобили в Байтландии

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

Ни для кого не секрет, что в Байтландии выпускается очень много различных моделей автомобилей. Хотя со временем устаревшие модели снимают с производства, им на смену приходит все больше новых. Каждая модель автомобиля характеризуется различными параметрами, и один из самых важных — стоимость автомобиля.

Футболисты местного футбольного клуба «Меткий баг» — довольно странные люди, и при покупке автомобиля в первую очередь руководствуются именно стоимостью автомобиля. Они не станут покупать слишком дорогую или дешевую модель. Футболисты считают модель слишком дорогой, если ее стоимость больше средней арифметической стоимости всех моделей, выпускаемых на данный момент, и слишком дешевой, если ее стоимость меньше. Модели со стоимостью, строго равной этому среднему арифметическому, напротив, пользуются большим спросом у футболистов. Компанию, выпускающую автомобили в Байтландии, заинтересовал вопрос: сколько моделей будут пользоваться спросом среди Байтландских футболистов в определенный момент?

Формат входного файла

В первой строке входного файла задано количество запросов n ($1 \leq n \leq 10^5$). В следующих n строках заданы запросы. Запросы бывают трех видов:

- запрос на добавление модели в производство в виде «+ k », где k — стоимость новой модели ($0 \leq k \leq 10^{13}$)
- запрос на удаление наиболее устаревшей модели из производства в виде «-». Наиболее устаревшей моделью считается та, которая начала выпускаться раньше, чем все остальные, выпускаемые в данный момент. Гарантируется, что при поступлении этого запроса в производстве есть хотя бы одна модель.
- запрос на вывод количества моделей в производстве, стоимость которых равна среднему арифметическому стоимостей всех автомобилей, выпускаемых в данный момент. Запрос поступает в виде строки «?». Гарантируется, что в момент запроса в производстве есть хотя бы одна модель.

Изначально в производстве нет ни одной модели.

Формат выходного файла

На каждый запрос вида «?» выведите ответ в отдельной строке. Ответы выводите в порядке следования запросов во входном файле.

Примеры

cars.in	cars.out
10	1
+ 1	0
+ 4	1
+ 3	
+ 4	
?	
-	
-	
?	
-	
?	

Задача D. Чемпионат

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

В футбольном чемпионате одной большой, но довольно холодной и северной страны, соревнуются между собой N команд. У каждой из них есть свое название, отражающее дух команды и ее историю. У каждой команды есть свой фан-клуб, состоящий из людей, следящих за всеми матчами, поддерживающих эту команду и в моменты грандиозных побед, и в моменты обидных поражений. Каждый из этих болельщиков в любой момент сможет назвать состав команды, результативность каждого игрока, соотношение забитых и пропущенных мячей, программу оставшихся до конца чемпионата матчей и результат любой из уже прошедших игр.

Естественно, что положение своей команды в текущей турнирной таблице каждый болельщик также знает наизусть. Более того, он помнит и положение всех остальных команд, и в любой момент сможет вам сказать, какая команда находится на третьем месте, а какая — на восьмом.

Однажды, проснувшись ранним утром после очередной победы своего любимого клуба, Иван понял, что совершенно не помнит турнирную таблицу. Все, что он помнит — это названия всех участвующих в чемпионате команд. После чашки крепкого кофе ему также удалось восстановить в памяти любопытный факт, замеченный им в пылу обсуждения таблицы с другими болельщиками. Он заметил, что строка, которая получается в результате последовательной записи названий всех команд в том порядке, в котором они следуют в таблице, лексикографически меньше, чем такая же строка, которая получилась бы при любом другом порядке записи названий команд. Так, если в чемпионате участвовали бы только команды «Dinamo-Kiev», «Dinamo-Moskva» и «Amkar», Иван бы с легкостью сказал, что «Amkar» занимает первое место, «Dinamo-Kiev» — второе, а «Dinamo-Moskva» — третье. Однако, в чемпионате участвуют другие команды, и Иван затруднился с восстановлением порядка их следования в таблице. Тогда он обратился за помощью к вам.

Формат входного файла

В первой строке входного файла дано одно целое число N ($1 \leq N \leq 10\,000$ — количество команд в чемпионате). В следующих n строках перечислены названия команд. Каждое название команды — это строка, состоящая только из строчных букв латинского алфавита. Длина названия одной команды не превышает 20 символов.

Формат выходного файла

В выходной файл выведите N строк, каждая из которых будет содержать номер команды, занимающей соответствующую позицию. Номера команд в ответе не должны повторяться. Если ответов несколько — выведите любой. Команды нумеруются с единицы в порядке их следования во входном файле.

Примеры

champ.in	champ.out
3	2
dinamom	3
amkar	1
dinamok	

Задача Е. Футбольные поля

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

Во Флатландии очень любят футбол. Обычные прямоугольные поля давно всем наскучили, так что во Флатландии придумали игру на полях произвольной формы. Форма поля — выпуклый многоугольник. По правилам Ассоциации Флатландского Футбола, каждый стадион должен иметь свою, оригинальную форму, не похожую на форму других стадионов.

Даны два многоугольника, задающих два поля. Вершины каждого многоугольника заданы в порядке обхода по или против часовой стрелки, начиная с произвольной вершины. Возможно, что многоугольники заданы разным порядком обхода. Вам необходимо определить, являются ли они одинаковыми. Два поля называются одинаковыми, если одно из другого можно получить некоторой последовательностью параллельных переносов, отражений относительно любой горизонтальной или вертикальной прямой, а также отражений относительно прямых $y = x$ или $y = -x$.

Формат входного файла

В первой строке входного файла задано число n ($3 \leq n \leq 10^5$) — количество вершин в многоугольнике. В следующих n строчках находится по два целых числа x и y ($-10^9 \leq x, y \leq 10^9$) — координаты очередной вершины первого многоугольника.

В следующих n строчках в таком же формате задаются вершины второго многоугольника.

Многоугольники, описанные во входном файле, выпуклые. Никакие три подряд идущие вершины одного многоугольника не лежат на одной прямой. Вершины каждого многоугольника заданы в порядке обхода по или против часовой стрелки, начиная с произвольной вершины. Возможно, что вершины многоугольников заданы в порядке разных обходов.

Формат выходного файла

В единственной строке выведите слово **YES**, если эти два поля одинаковы, и **NO** иначе.

Примеры

<code>fields.in</code>	<code>fields.out</code>
3 0 0 1 1 1 0 2 2 3 2 3 1	YES

Задача F. Мы - мясо! Мы - газ!

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

Футбольный клуб «Газмяс» в этом сезоне тренирует новый тренер, поскольку методами предыдущего тренера руководство и игроки были слегка не довольны. Новый тренер подходит ко всем вопросам с математической точки зрения. Если раньше перед тренировкой игроки собирались в кружок и слушали тренера, то у нового тренера другой подход. Он строит игроков в линию в алфавитном порядке. А так как тренер окончил Институт Точного Математического Образования, он легко может с точностью до сантиметра определить рост каждого игрока.

Далее задача тренера состоит в том, чтобы определить состав на следующую игру. Для этого он хочет определить сколько у него есть пар сыгранных футболистов. Но у нового тренера понятие «Пара сыгранных футболистов» очень странно. Два футболиста считаются сыгранными, если при построении в алфавитном порядке игрок с большим ростом стоит левее, чем второй футболист, рост обоих футболистов четен, и между ними стоит хотя бы один футболист с нечетным ростом.

Хотя новый тренер и очень умен, но в этом сезоне состав «Газмяса» очень большой (оно и понятно, «Газмяс» собирается выйти в Премьер-лигу). Поэтому он не может сам посчитать количество пар и просит Вас помочь ему. Перед тем, как вы приступили к выполнению этого ответственного задания, тренер сообщил Вам, что рост всех игроков различен.

Формат входного файла

Первая строка содержит одно число N ($1 \leq N \leq 10^5$) — количество футболистов. Вторая строка содержит N чисел, идущих через пробел — рост футболистов. Рост каждого футболиста положителен и не превышает N . Рост всех футболистов различен. Футболисты уже упорядочены по алфавиту.

Формат выходного файла

Выведите одно число — количество пар сыгранных футболистов.

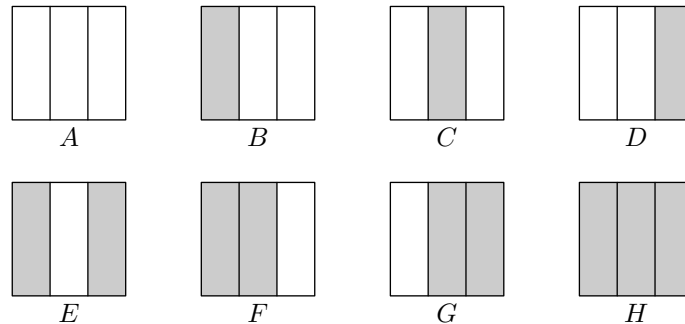
Примеры

<code>gazmyas.in</code>	<code>gazmyas.out</code>
5 4 3 2 5 1	1
5 4 2 3 5 1	0

Задача G. Дженга

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

После того как Александру и Игорю на работе не повысили зарплату, они уволились и увлеклись дженгой. В игре дженга есть башенка состоящая из нескольких слоев. Каждый слой состоит из не более чем трех расположенных параллельно блоков. Блоки из соседних слоев лежат перпендикулярно друг другу. Ниже нарисованы все возможные слои, где серый цвет соответствует наличию блока, а белый — его отсутствию.



Слои *A*, *B*, *D* - нестабильные, остальные стабильные. В начальный момент времени все слои башенки стабильные за исключением, возможно, верхнего. Верхний может быть нестабильным. Александр и Игорь ходят поочередно и Александр ходит первым. За один ход игрок берет блок из любого слоя и кладет его либо в верхний слой, если там есть свободные позиции, либо создает новый верхний слой. При этом, если блок был взят из последнего слоя, то его нельзя положить в этот же слой, а необходимо создать новый. Если после хода игрока существует нестабильный слой кроме самого верхнего, то он проиграл. Определите, кто проиграет при оптимальной игре обоих игроков.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leq n \leq 700$) — количество слоев в башенке в начальный момент. В следующих n строчках дано описание слоев i -тая из которых содержит три числа от 0 до 1, где 0 - соответствует отсутствию блока на соответствующей позиции слоя, а 1 - его наличию.

Формат выходного файла

Если победит Александр выведете `Alex`, иначе `Igor`.

Примеры

<code>jenga.in</code>	<code>jenga.out</code>
2 1 1 1 1 1 1	Igor
1 1 1 1	Alex

Задача Н. Лампочка

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

У Эдуарда Константиновича очень важная профессия — он следит за тем, чтобы, во время футбольных матчей, весь стадион был хорошо освещен. Он долго обучался этому и теперь считает себя одним из самых квалифицированных людей в этой области. Вы не поверите, но Эдуард Константинович даже каждый год ездит на конференции, которые посвящены правильному освещению футбольных полей. Но однажды ему приснился сон, который потом долго не давал ему покоя. . .

В этом сне он попал в «квадратный» мир. Он был до боли похож на обычный, но все же в нем были некоторые отличия. Например, в «квадратном» мире расстояние между точками (x_1, y_1) и (x_2, y_2) в некоторой системе отсчета считается равным $|x_2 - x_1| + |y_2 - y_1|$, а не таким, каким он привык его считать. Во сне он был обладателем той же самой профессии. Ему нужно было осветить футбольное поле.

В мире, в который попал Эдуард Константинович, футбольное поле представляет собой прямоугольник, разделенный на квадратные клетки шириной в один метр — w клеток по горизонтали, h по вертикали. Над некоторыми из этих клеток висят лампочки, которые и освещают поле во время матча. Темнотой некоторой клетки считается расстояние до ближайшей лампочки (заметим, что клетка с темнотой 0 является самой освещенной). Качество работы Эдуарда Константиновича определяется клеткой с наименьшей освещенностью. Чем больше темнота этой клетки, тем хуже была проведена работа. Кроме тех n лампочек, которые уже висят над полем, у Эдуарда Константиновича была в запасе еще одна, которую он мог повесить над какой-нибудь клеткой.

Почему же этот сон не давал ему покоя? Потому что вариантов, куда повесить лампочку, было слишком много (нетрудно заметить, что всего их $W \cdot H$) и он так и не смог определиться. Теперь он усомнился в том, что является одним из лучших специалистов в мире по освещению футбольных полей. Вам необходимо помочь Эдуарду Константиновичу понять, куда же нужно было повесить лампочку.

Формат входного файла

В первой строке входного файла даны три целых числа W, H, n ($1 \leq W, H \leq 1000, 1 \leq n \leq 10^5$) — стороны футбольного поля и количество лампочек. В каждой из следующих n строк находится два целых числа x и y ($1 \leq x \leq W, 1 \leq y \leq H$) — положение очередной лампочки.

Формат выходного файла

В первой строчке выведите самую большую темноту (расстояние до ближайшей лампочки) среди всех клеток поля. Во второй строке выведите координаты лампочки — два целых числа x и y ($1 \leq x \leq W, 1 \leq y \leq H$). Если существует несколько вариантов повесить лампочку так, чтобы самая темная клетка была как можно более освещенной, разрешается вывести любой.

Примеры

<code>lights.in</code>	<code>lights.out</code>
3 3 2	1
2 1	2 2
2 3	

Задача I. Стадион

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

В Берляндии скоро пройдет чемпионат мира по футболу, и сейчас активно ведется его подготовка. Для проведения особо важных матчей и церемоний открытия и закрытия планируется построить новый стадион, который должен стать самым большим из известных человечеству. Однако, если стадион будет очень большим, то зрители могут рассаживаться очень долго. Для решения этой проблемы был создан специальный отдел оптимизации стадиона.

По проекту, зрительские места на стадионе будут разбиты на несколько одинаковых секторов. Сектор представляет из себя несколько рядов кресел по m кресел в каждом ряду с двумя проходами по бокам, так что к каждому месту можно будет пройти справа или слева. В каждом ряду кресла пронумерованы слева направо. Способом рассадки зрителей в одном ряду отдел называет перестановку чисел от единицы до m , соответствующую порядку, в котором зрители занимают свои места в этом ряду. Хорошим способом назовем такой способ, при котором никакому зрителю не придется проходить на пути к своему месту мимо уже сидящего человека. Вас же попросили посчитать, сколько существует различных таких способов. Считается, что очередной зритель начинает идти к своему месту только тогда, когда предыдущий за ним (если такой был) уже занял свое место.

Формат входного файла

В входном файле содержится одно целое число m ($1 \leq m \leq 10^{18}$) — число мест в каждом ряду.

Формат выходного файла

В выходной файл выведите одно число — число различных хороших способов рассадки зрителей по модулю $10^9 + 7$.

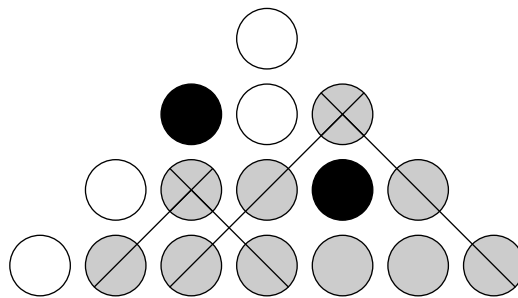
	<code>stadium.in</code>	<code>stadium.out</code>
1	1	1
3	3	4

Задача J. Съёмка

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

До финального матча чемпионата Берляндии остались считанные дни! Но, так как всё делается в последний момент, некоторые организационные моменты ещё не улажены. В том числе нужно организовать съёмку матча. В городе есть несколько компаний, которые готовы организовать съёмку. Но они все конкурируют, и хотят снимать матч, а не операторов конкурентных компаний!

Операторов можно разместить на трибунах стадиона. Трибуны состоят из n рядов: в первом $2n-1$ место, во втором — $2n-3, \dots$, в n -м — одно. Однако, не все подходят к матчу безответственно, и некоторые болельщики уже купили себе билеты на определённые места. Соответственно, операторов можно поставить только на свободные.



Пример стадиона. Серым отмечены места, которые снимаются операторами. Чёрным — занятые зрителями места. Операторы отмечены крестиками. Для каждого из них показано, что снимает его камера.

Но конкуренция есть не везде, и все операторы используют камеры одной и той же модели, неизменяемый угол обзора которой позволяет снимать три места в следующем ряду, пять — через ряд, итд. Власти хотят, чтобы всё прошло на высшем уровне, и попросили предоставить им возможные способы расставить операторов, чтобы самим выбрать наиболее оптимальный. Однако, начальство стадиона не считает это хорошей идеей, полагая, что таких способов очень много. Помогите им выяснить, сколько же существует способов.

Формат входного файла

В первой строке заданы числа n и m ($1 \leq n \leq 2000$, $0 \leq m \leq 1000$) — число рядов и количество занятых мест.

Далее в m строках заданы занятые места по одному в строке. Место задается двумя числами: r и p ($1 \leq r \leq n$, $1 \leq p \leq 2(n-r)+1$) — номер ряда и номер места в этом ряду.

Формат выходного файла

Выведите остаток от деления количества способов на $10^9 + 7$.

Примеры

<code>operators.in</code>	<code>operators.out</code>
2 1 1 2	5