Задача A. Arbitrage

Имя входного файла:	`arbitrage.in`
Имя выходного файла:	`arbitrage.out`
Ограничение по времени:	3 seconds
Ограничение по памяти:	256 мегабайта

Сергей работает в крупной Антарктической Финансовой Компании. Недавно он получил доступ к межвалютному обменному каналу, используя который, иногда можно обменять валюту по лучшему курсу, чем соотношение официальных курсов соответсвующих валют.

Сергей решил провести несколько дополнительных обменных операций с проходящими через него суммами и получить с них дополнительный доход.

При валютообменных операциях в АФК все вычисления ведутся в Антарктических Долларах (АД) по официальному курсу; к примеру, обмен a тугриков на b евро при курсах C_t и C_e соответственно представляется как операция, при которой списывается $a \cdot C_t$ АД — эквивалент обмениваемой суммы, а зачисляется $b \cdot C_e$ АД — эквивалент получаемой суммы.

Обмен по межвалютному каналу возможен только для некоторых пар валют (i, j) (при этом наличие возможности обмена i-й валюты на j-ю не гарантирует наличие возможности обратного обмена). Для каждой такой пары известна прибыль $a_{i,j}$, получаемая на каждые 1000 АД эквивалента обмениваемой суммы (если это значение отрицательно, то операция обмена убыточна), а также лимит обмениваемой одним пользователем суммы $c_{i,j}$, заданный в тысячах АД. Интерфейс системы предоставляет возможности только для работы с суммами, эквивалент которых кратен 1000 АД.

После исследования различных возможностей Сергеем был разработан следующий план действий.

Для некоторых пар валют (i, j) проводятся обмены через "специальный" канал так, чтобы в начале и в конце последовательности обменов сумма (эквивалент) каждой из валют была одинакова. Полученная в результате промежуточных действий прибыль переводится на счёт Сергея.

Вычислите, какой максимальный доход может получить Сергей по завершении последовательности обменов, считая, что он может задействовать в своих операциях неограниченное количество каждой из валют.

Формат входного файла

В первой строке входного файла заданы целые числа n — количество различных иностранных валют, с которыми работает АФК ($2 \le n \le 30$) и m ($1 \le m \le n \cdot (n-1)$) — количество возможных направлений обмена, предоставляемых по межвалютному каналу.

Каждая из последующих m строк описывает одно направление обмена. Направление задаётся четырьмя целыми числами: идентификатором исходной валюты i, идентификатором j валюты, в которую происходит обмен, прибыль в АД за каждые 1000 АД эквивалента обмениваемой суммы $a_{i,j}$ ($-100 \le a_{i,j} \le 100$, $a_{i,j}$ — целое) и максимальным количеством валюты, которое может быть обменяно одним пользователем (в тысячах АД) $c_{i,j}$ ($1 \le c_{i,j} \le 100$, $c_{i,j}$ — целое).

Формат выходного файла

Выведите одно целое число — максимально возможную прибыль в АД, которую Сергей может получить в результате реализации своего плана.

Пример

arbitrage.in	arbitrage.out
4 6 1 2 3 2 2 1 -1 1 2 3 0 1 3 4 0 2 4 1 2 2 1 3 -3 1	7

Задача B. Border (Division 1 Only!)

Имя входного файла:	`border.in`
Имя выходного файла:	`border.out`
Ограничение по времени:	4 seconds
Ограничение по памяти:	256 мегабайта

Колонисты с Земли собираются устанавливать новое силовое поле на Пандоре. Они хотят закрыть этим полем свою. базу и некоторую прилегающую к ней область.

База представляет собой треугольник со сторонами, равными a, b и c соответственно. Мощность установленного на базе генератора позволяет создать силовое поле — замкнутую кривую длины l. Колонисты хотят установить поле так, чтобы, во-первых, база полностью находилась внутри поля, а, во-вторых, чтобы поле дополнительно защищало как можно больше территории.

Вычислите общую площадь, которую защищает установленное в соответствии с требованиями колонистов силовое поле.

Формат входного файла

Во входном файле заданы четыре целых числа a, b, c и l ($1 \le a, b, c \le 100$, $a + b + c \le l \le 2000$, a, b и c являются сторонами невырожденного треугольника).

Формат выходного файла

Выведите одно вещественное число — общую площадь, защищённую установленным в соответствии с требованиями колонистов силовым полем. Абсолютная или относительная ошибка при выводе площади не должна превышать 10^{-6}.

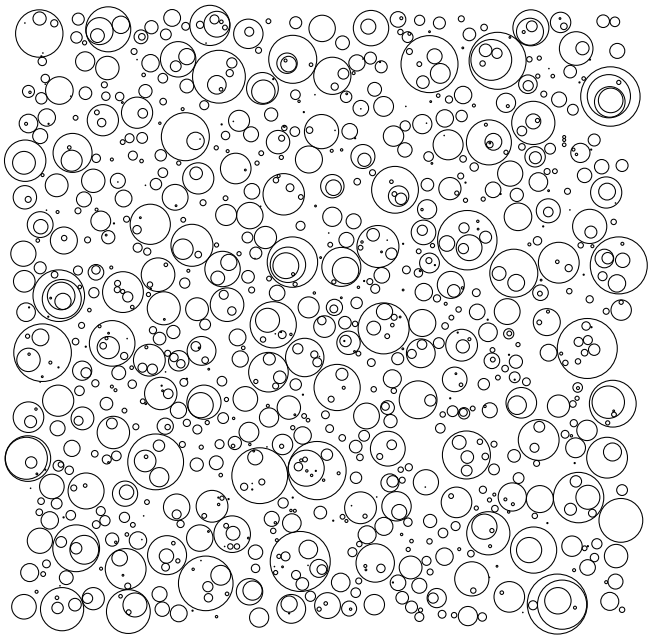
Пример

border.in	border.out
3 4 5 100	795.774715459476679
3 4 5 12	6.0
3 4 5 13	12.261818799960896

Задача C. Circles (Division 1 Only!)

Имя входного файла:	`circles.in`
Имя выходного файла:	`circles.out`
Ограничение по времени:	7 seconds
Ограничение по памяти:	256 мегабайта

На плоскости заданы n кругов. Круги могут пересекаться, однако для любых двух кругов их пересечением являетя или одна точка, или один из этих двух кругов.



Найдите площадь, покрытую объединением этих кругов.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 100\,000$). Каждая из последующих n строк содержит по три целых числа и задаёт один круг. В i-й строке задаётся i-й круг координатами центра x_i и y_i и радиусом r_i ($-10^6 \le x_i, y_i \le 10^6$, $1 \le r_i \le 10^6$).

Формат выходного файла

Выведите одно вещественное число: площадь объединения этих кругов (то есть множества точек, покрытых хотя бы одним кругом). При этом абсолютная или относительная ошибка не должна превышать 10^{-9}.

Пример

circles.in	circles.out
4 2 2 2 2 2 1 5 2 1 5 5 2	28.2743338823081391

Задача D. Diamonds and Golden Strings

Имя входного файла:	`diamonds.in`
Имя выходного файла:	`diamonds.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 мегабайта

Древняя ацтекская цивилизация знаменита своими экзотическими ритуалами. Не так давно археологи, раскапывая древние руины, обнаружили описание следующей игры, популярной среди ацтекских аристократов.

На стол кладётся несколько золотых ожерелий с алмазами. Игроки делают ходы по очереди.

Каждое ожерелье представляет собой замкнутую цепочку из трёх или более алмазов, в которой соседние алмазы соединены между собой золотой нитью. Игрок, делающий ход, перерезает одну из нитей. Если после перерезания нити какой-то алмаз не соединён ни с какими другими алмазами, игрок забирает этот алмаз и продолжает ход, перерезая другую нить. Если после перерезания нити каждый алмаз остаётся соединённым как минимум с одним другим алмазом, ход переходит к другому игроку.

Игра заканчивается, когда взят последний алмаз. Цель игры — забрать наибольшее количество алмазов.

В найденном археологами описании игра начиналась с n ожерелий, содержащих $a_1, a_2, \ldots, a_n$ алмазов соответственно. Выясните, сколько алмазов забирает первый игрок и сколько — второй в случае, когда оба действуют оптимально.

Формат входного файла

Первая строка входного файла содержит целое число n — количество ожерелий на столе в начале игры ($1 \le n \le 100$). Вторая и последняя строка содержит n целых чисел — количество алмазов в ожерельях $a_1, a_2, \ldots, a_n$ ($3 \le a_i \le 10^9$).

Формат выходного файла

Выведите два целых числа — количество алмазов, которое при оптимальной игре возьмёт игрок, делающий ход первым и игрок, делающий ход вторым, соответственно.

Пример

diamonds.in	diamonds.out
2 3 3	3 3
2 5 5	4 6

Во втором примере оптимальная игра выглядит так. Первый игрок разрезает любую нить. Перед ходом второго игрока есть цепочка длины 5 и замкнутое ожерелье длины 5. Он отрезает и забирает один алмаз от цепочки, после чего разрезает оставшуюся цепочку длины 4 на две цепочки длины 2. Первый игрок в любом случае вынужден разомкнуть второе ожерелье, так что он перед этим разрезает обе цепочки длины 2, забирая 4 алмаза. После того, как второе ожерелье будет разомкнуто, ход перейдёт ко второму игроку, который забирает себе все 5 оставшихся алмазов, отрезая их один за другим.

Задача E. Expedition to Mars

Имя входного файла:	`expedition.in`
Имя выходного файла:	`expedition.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 мегабайта

Колонисты с Земли планируют построить на Марсе новую базу. Марсианская база представляет собой n одинаковых кубических контейнеров, размещённых на поверхности планеты.

При размещении контейнеров требуется соблюдать следующие правила:

- Контейнеры не разрешается ставить друг на друга, так что одна грань каждого контейнера должна прилегать к поверхности Марса.
- Контейнеры должны быть расположены так, чтобы их стороны были параллельны направлениям с севера на юг и с запада на восток.
- Если два контейнера соприкасаются, то они имеют общую грань или общее ребро.
- Так как внешние (не являющиеся общими ни для каких двух контейнеров) грани требуют дополнительной защиты от атмосферы Марса, то контейнеры должны быть размещены так, чтобы суммарная площадь таких граней была минимально возможной

Руководитель экспедиции хочет узнать, сколькими способами можно расположить контейнеры в соответствии с вышеперечисленными правилами. Так как число способов может быть весьма велико, выведите его по модулю $10^9 + 7$.

Формат входного файла

Во входном файле задано одно целое число — количество контейнеров ($1 \le n \le 500$).

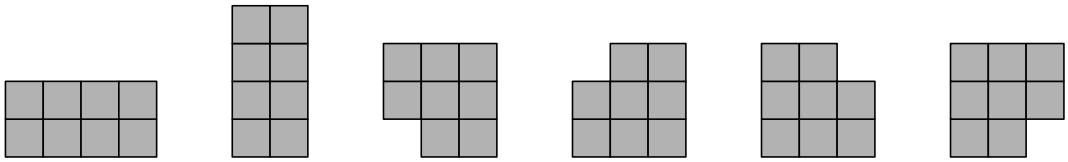
Формат выходного файла

Выведите одно целое число — количество способов постройки базы в соответствии с вышеперечисленными правилами, взятое по модулю $10^9 + 7$.

Пример

expedition.in	expedition.out
8	6

На иллюстрации показаны все 6 возможных способов постройки базы.



Задача F. Formula Verification (Division 1 Only!)

Имя входного файла:	`formula.in`
Имя выходного файла:	`formula.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 мегабайта

Известно, что логики первого порядка над целыми числами являются неразрешимыми. Не существует алгоритма, который бы мог по формуле с кванторами $\forall$ и $\exists$, умножением, сложением и операторами отношения определить, является ли данная формула верной для всех целых чисел.

В отличие от целочисленного случая, логика первого порядка над вещественными числами является разрешимой. В этой задаче Вам предстоит проверить, является ли логическая формула первого порядка с двумя переменными и двумя кванторами истинной на множестве вещественных чисел.

В данной задаче формулы определяются следующим образом:

$$
\begin{aligned}
\langle \text{formula} \rangle &\longrightarrow \langle \text{quantifier} \rangle \texttt{ x } \langle \text{quantifier} \rangle \texttt{ y } \langle \text{polynomial} \rangle \langle \text{relation} \rangle \langle \text{polynomial} \rangle \\
\langle \text{quantifier} \rangle &\longrightarrow \forall \mid \exists \\
\langle \text{relation} \rangle &\longrightarrow \texttt{=} \mid \texttt{<} \mid \texttt{>} \mid \texttt{<=} \mid \texttt{>=} \\
\langle \text{polynomial} \rangle &\longrightarrow \langle \text{term} \rangle \mid \langle \text{polynomial} \rangle + \langle \text{term} \rangle \mid \langle \text{polynomial} \rangle - \langle \text{term} \rangle \\
\langle \text{term} \rangle &\longrightarrow \langle \text{factor} \rangle \mid \langle \text{term} \rangle * \langle \text{factor} \rangle \\
\langle \text{factor} \rangle &\longrightarrow x \mid y \mid \langle \text{integer} \rangle \\
\langle \text{integer} \rangle &\longrightarrow \textit{integer from 0 to 100}
\end{aligned}
$$

В определении выше "⟨polynomial⟩" — многочлен от переменных x и y степени не выше второй.

Формула $\forall a F(a)$ истинна, если для всех вещественных чисел a предикат $F(a)$ является истинным. Формула $\exists a F(a)$ истинна, если хотя бы для одного вещественного числа a предикат $F(a)$ является истинным.

Примеры истинных формул: "$\forall x \exists y\, x < y$", "$\forall x \exists y\, x * x = y$", "$\forall x \forall y\, x * x + y * y \geq 0$". Примеры ложных формул: "$\exists x \forall y\, x < y$", "$\forall x \exists y\, x = y * y$".

Формат входного файла

Входной файл состоит из не более чем 20 тестовых примеров, при этом каждый тестовый пример задан в отдельной строке.

Каждая строка входного файла содержит формулу первого порядка, соответствующую описанию, приведённому в задаче. Формула использует переменные "`x`" и "`y`", операторы "`*`", "`+`", "`-`", операторы отношения "`>`", "`<`", "`=`", "`>=`", "`<=`", кванторы "`A`" для $\forall$, и "`E`" для $\exists$, а также неотрицательные числа, не превосходящие 100.

Пробелы могут встречаться в произвольном месте формул, за исключением ситуации, когда пробел разрывает целое число или знак отношения. Гарантируется, что при приведении заданных полиномов к виду, при котором не существует двух различных термов с одинаковыми степенями и x, и y коэффициенты не будут превышать 1000, в том числе и коэффициенты, которые появляются при адекватных способах промежуточных вычислений. Длина каждой из строк не превосходит 200.

Формат выходного файла

Для каждой формулы выведите в отдельной строке "`true`", если формула истинна над множеством целых чисел, и "`false`" в противном случае.

Пример

formula.in	formula.out
AxEy y=x*x AxEy x=y*y	true false

Задача G. Great Minds (Division 1 Only!)

Имя входного файла:	great.in
Имя выходного файла:	great.out
Ограничение по времени:	11 seconds
Ограничение по памяти:	256 мегабайта

Для телевизионной игре "Great Minds" отбираются n вопросов. Отобранные вопросы пронумерованы в соответствии с их сложностью от 1 до n, при этом никакие два вопроса не имеют одинаковой сложности.

Перед началом игры вопросы выкладываются в ряд на игровой стол. При этом вопрос, лежащий i-м по порядку, имеет сложность a_i.

В игре принимает участие команда из трёх игроков. В начале игры каждый игрок выбирает вопрос, при этом никакие два игрока не могут выбрать один и тот же вопрос.

Далее участники пытаются отвечать на вопросы. Если игрок ответил на вопрос, расположенный j-м по порядку, правильно, он может или выйти из игры, или выбрать новый вопрос, расположенный далее, чем j-й (то есть с порядковым номером, большим j), сложность которого больше сложности j-го вопроса.

Если в результате игры не останется вопросов, на которые не ответил ни один участник, то команда выигрывает приз.

Участники заметили, что при некоторой изначальной расстановке вопросов приз невозможно выиграть вне зависимости от уровня игры команды и выбранной командой стратегии — например, если вопросы расположены в порядке $\langle 4, 3, 2, 1\rangle$, то ответить на все вопросы невозможно: ни для какого вопроса не существует вопроса с большим номером и большей сложностью одновременно, так что, кроме трёх, выбранных первоначально, больше ни один вопрос отыгран не будет.

Игроков интересует, сколько существует различных начальных расположений вопросов, при которых выигрыш приза возможен. Так как результат может быть достаточно велик, вычислите его по модулю $10^9 + 7$.

Формат входного файла

Во входном файле содержится одно целое число n ($3 \le n \le 500$).

Формат выходного файла

Выведите одно целое число — количество начальных расположений вопросов, позволяющих выиграть приз, взятое по модулю $10^9 + 7$.

Пример

great.in	great.out
4	23

Задача H. Highways (Division 1 Only!)

Имя входного файла:	`highways.in`
Имя выходного файла:	`highways.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 мегабайта

Изучая проект бюджета на очередной год, король Флатландии заметил, что на эксплуатацию хайвэев запланированы слишклом большие траты. Король вызвал министра транспорта и поинтересовался, не является ли дорожная сеть страны избыточной.

В Флатландии n крупных городов, соединённых m двунаправленными хайвэями. Король выбрал число k и определил, что система хайвэев является избыточной, если существует множество A, содержащее s городов, такое, что между входящими в A городами построено более, чем $k(s-1)$ хайвэев.

Министр утверждает, что система хайвэев не является избыточной, однако король не принял это утверждение министра на веру и поручил Вам проверить, действительно ли это так.

Формат входного файла

В первой строке входного файла заданы три целых числа n, m и k — соответственно количетво городов, количество хайвэев и уровень избыточности ($2 \le n \le 100$, $1 \le m \le n(n-1)/2$, $1 \le k \le 20$). Каждая из последующих m строк описывает хайвэй и содержит два числа — номера городов, который соединяет данный хайвэй. Города занумерованы от 1 до n, каждые два города соединены не более, чем одним хайвэем, не существует хайвэев, соединяющих какой-либо город с самим собой.

Формат выходного файла

Если система хайвэев не является избыточной, выведите "`OK`" в первой строке выходного файла.

Если система хайвэев является избыточной, выведите "`Redundant`" в первой строке выходного файла. Далее выведите описание множества A, которое удостоверяет избыточность системы. Описание начинается с числа s — количества городов в множестве A, за которым следует s целых чисел — номера городов в A. Если имеется несколько таких множеств, выведите любое из них.

Пример

highways.in	highways.out
5 8 2 1 2 1 3 2 3 3 5 3 4 5 2 2 4 4 5	OK
5 5 1 1 2 2 3 3 4 4 5 5 1	Redundant 5 1 2 3 4 5

Задача I. Intercity Express (Division 1 Only!)

Имя входного файла:	`intercity.in`
Имя выходного файла:	`intercity.out`
Ограничение по времени:	8 seconds
Ограничение по памяти:	256 мегабайта

Андрей разрабатывает справочную систему, информирующую о наличии билетов на поезда.

Тестируется система на примере Intercity Express line, соединяющей два крупных города и содержащей $n-2$ промежуточных станций; таким образом, маршрут поезда состоит из n станций, занумерованных целыми числами от 1 до n. Места в поезде занумерованы от 1 до s.

При тестировании система получает доступ к базе проданных билетов на один рейс этого поезда в направлении от станции 1 до станции n. Система должна отвечать на запросы следующего типа: есть ли свободные (то есть не проданные ни для одного участка пути на маршруте от a до b) места от станции a до станции b, и, если есть, выводить наименьший номер такого места.

Для того, чтобы помочь организовать тестирование, напишите программу, которая будет отвечать на подобные запросы, поддерживая внутренний формат системы.

Формат входного файла

Первая строка входного файла содержит n — количество станций, s — количество мест в поезде и m — количество уже проданных мест ($2 \le n \le 10^9$, $1 \le s \le 100\,000$, $0 \le m \le 100\,000$).

Последующие m строк описывают проданные билеты. Каждый билет задаётся тремя целыми числами c_i, a_i, и b_i — номером проданного места, станцией отправления и станцией прибытия соответственно ($1 \le c_i \le s$, $1 \le a_i < b_i \le n$).

Следующая строка содержит q — количество запросов ($1 \le q \le 100\,000$).

Для совместимости с существующей базой данных при чтении запросов должна поддерживаться специальная переменная p. Изначально $p = 0$. Последующие $2q$ целых чисел описывают запросы. Каждый запрос задаётся двумя целыми числами: x_i и y_i ($x_i < y_i$). Чтобы получить номера городов, возможность проезда между которыми запрашивается, используйте следующие формулы: $a = x_i + p$, $b = y_i + p$. Ответом на запрос будет 0, если свободных мест для проезда между станциями a и b нет, или наименьший номер места, которое свободно на всём протяжении пути от a до b.

После ответа на очередной запрос значение этого ответа присваивается переменной p.

Формат выходного файла

Для каждого запроса выведите ответ на этот запрос.

Пример

intercity.in	intercity.out
5 3 5 1 2 5 2 1 2 2 4 5 3 2 3 3 3 4 10 1 2 1 2 1 2 2 3 -2 0 2 4 1 3 1 4 2 5 1 5	1 2 2 3 0 2 0 0 0 0

В данном примере номера мест в запросах — $(1,2)$, $(2,3)$, $(3,4)$, $(4,5)$, $(1,3)$, $(2,4)$, $(3,5)$, $(1,4)$, $(2,5)$, $(1,5)$.

Задача J. Jungle Speed

Имя входного файла:	`jungle.in`
Имя выходного файла:	`jungle.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 мегабайта

Jungle Speed — карточная игра на базе нестандартного набора карт, разработанная компанией Asmodee Editions. В данной задаче используется упрощённая версия игры, при этом различные типы карт обозначены различными латинскими заглавными буквами.

Колода содержимт по две карты каждого типа. Изначально все карты распределены между участниками, каждый участник кладёт свои карты перед собой в стопку "рубашкой" вверх. Эти карты называются "колодой" участника. В течение игры некоторые карты из "колоды" участника складываются "рубашкой" вниз перед участником в другую стопку, называемую "сбросом".

Игроки делают ходы по часовой стрелке, начиная с самого младшего. Пронумеруем участников в соответствующем порядке номерами от 1 до n.

Во время хода участник берёт самую верхнюю карту из колоды, переворачивает её "лицом" вверх и кладёт её наверх "сброса". Если после этого наверху "сбросов" каких-то двух участников оказываются одинаковые карты, эти участники "вступают в поединок"

Поединок заключается в следующем — участники должны как можно быстрее схватить некий предмет — "тотем", лежащий на середине стола. Тот, кто прореагировал медленнее, забирает в "колоду" карты из своего "сброса" и "сброса" участника, выигравшего поединок. Сначала перекладываются карты из "сброса" проигравшего участника, начиная с самой верхней карты, затем — карты из "сброса" выигравшего, также начиная с самой верхней карты. Каждая новая карта добавляется в "колоду" снизу уже существующих.

Если у игрока нет карт в "колоде", ход передаётся следующему по часовой стрелке игроку, это действие производится автоматически и ходом не считается.

Игра завершается в случае, если выполнено одно из следующих условий.

- У некоторого игрока нет карт ни в "колоде", ни в "сбросе". Этот игрок объявляется победителем. В случае, если таких игроков оказывается несколько, победителем объявляется тот, у кого раньше закончилась "колода".
- Все карты находятся в "сбросах", и никакие две верхние карты "сбросов" не совпадают. В этом случае игра заканчивается вничью.
- Если игра продолжается 10^6 ходов и у всех игроков есть карты или в "колоде", или в "сбросе", при этом как минимум у одного игрока есть карта в "колоде", игра заканчивается по времени.

Вам заданы изначальные "колоды" всех n участников, а также скорость их реакции. Выясните результат игры — будет ли игра закончена по времени, сведена вничью или же завершится победой одного из участников.

Формат входного файла

Первая строка входного файла содержит n — количество игроков ($2 \le n \le 20$). Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ — время реакции игроков ($1 \le a_i \le 100$). Если два игрока вступают в поединок, выигрывает игрок с меньшим временем реакции. Если два игрока с одинаковой реакцией вступают в поединок, тот, кто открывал карту последним, проигрывает.

Последующие n строк содержат последовательность заглавных латинских букв и описывают "колоды" игроков, заданные в порядке сверху вниз. Изначально каждый игрок имеет не менее одной карты в "колоде". Каждая буква встречается не более во входном файле не более двух раз.

Формат выходного файла

Если игра завершилась вничью, выведите "`Draw after X moves.`", где X — количество ходов, сделанных до того, как все карты оказались в "сбросах". Напоминаем, что ситуация, в которой игрок пропускает ход из-за отсутствия карт в "колоде" ходом не считается.

Если некоторый игрок выигрывает, выведите "`Player Y wins after X moves.`", где Y — номер игрока, который выиграл, а X — количество ходов, которое прошло с момента начала игры (включая выигрышный ход игрока Y).

Если игра заканчивается по времени после 10^6 ходов, выведите "`Abandoned after 1000000 moves.`".

Пример

jungle.in	jungle.out
2 1 2 ABC CBA	Player 1 wins after 8 moves.
2 2 1 ABC CBA	Player 2 wins after 8 moves.
2 2 1 ABCD CDAB	Draw after 8 moves.

Задача K. Investments (Division 2 Only!)

Имя входного файла:	`invest.in`
Имя выходного файла:	`invest.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

Профессор Громов разработал проект нового робота. Однако для производства первой партии роботов необходима сумма денег, которая у профессора отсутствует, так что профессор вынужден привлекать инвестиции, чтобы покрыть расходы на выпуск первой партии.

Профессору известно, сколько компаний готовы поддержать его новую разработку и сколько денег они готовы в неё инвестировать. Профессор хочет привлечь к работе над проектом как можно меньше компаний, а если расходы всех инвесторов не в состоянии покрыть стоимость выпуска первой партии — отказаться от проекта как от малоэффективного.

Помогите профессору вычислить, какое наименьшее количество компаний он должен привлечь для финансирования нового проекта.

Формат входного файла

Первая строка содержит целое число T — количество тестовых примеров ($1 \le T \le 255$). Каждый тестовый пример описывает новый проект. Первая строка описания содержит два целых числа: стоимость проекта N ($1 \le N \le 10^6$), которую надо покрыть за счёт инвестиций и количество потенциальных инвесторов C ($1 \le C \le 1000$). Вторая строка содержит C целых чисел, i-е из которых задаёт максимальную сумму M_i, которую i-я компания готова вложить в проект.

Формат выходного файла

Для каждого тестового примера выведите одну строку — наименьшее количество инвесторов, которое следует привлечь профессору Громову для того, чтобы начать производство. Если денег не хватит ни при каком количестве инвесторов, выведите "`impossible`".

Примеры

invest.in	invest.out
3 101 6 23 27 41 19 23 57 100 7 20 22 33 8 24 67 9 1001 3 123 456 123	3 2 impossible

Задача L. Modules (Division 2 Only!)

Имя входного файла:	`modules.in`
Имя выходного файла:	`modules.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

Для завершения работы над новой моделью робота профессору Громов надо настроить несколько модулей — например, модуль ориентации в пространств или модуль распознавания текстовой информации. Эта работа производится внешними компаниями.

Однако цена, которую компания выставляет за работу по настройке предоставленного ею модуля, зависит от степени готовности робота — например, если система позиционирования в пространстве уже готова, то специалист по настройке интерфейса запросит большую сумму, так как он вынужден настраивать дополнительные функции, и так далее. То есть наценки зависят от того, какие модули уже настроены, и профессор хочет определить порядок работ так, чтобы затратить наименьшую сумму.

Модули (и работы по их настройке) пронумерованы целыми числами от 1 до n. По заданной базовой стоимости каждой работы p и значениям наценок s для каждой пары работ (i, j) при $i \neq j$, обозначающих, что наценка s за работу i платится тогда и только тогда, когда работа j была проведена перед работой i, вычислите наименьшую сумму, которую необходимо затратить для завершения всех работ.

Формат входного файла

Первая строка входного файла содержит цеоле число n, $1 \leq n \leq 14$ — количество модулей (и работ по их настройке). Далее следуют n строк, каждая из которых содержит ровно n целых чисел. i-я строка содержат цену за i-ю работу, а также наценки за неё в следующем формате: для $j \neq i$ j-е число в i-й строке описывает наценку на i-ю работу, начисляемую в случае, если j-я работа была сделана до неё. i-е число в i-й строке описывает базовую цену за i-ю работу. Все цены — неотрицательные целые числа, не превосходящие 10^5.

Формат выходного файла

Выведите единственную строку, содержащую целое число p — требуемую минимальную сумму.

Пример

modules.in	modules.out
2 13 13 900 13	39
3 12 24 0 0 12 0 1234 5678 12	36

Задача M. OCR (Division 2 Only!)

Имя входного файла:	`ocr.in`
Имя выходного файла:	`ocr.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

Для нового поколения роботов профессора Громова компания-поставщик разработала улучшенный модуль распознавания символов (OCR). Подобные модули используется при чтении роботом "бумажного" текста. К сожалению, распознавание не всегда работает успешно, так что некоторые символы остаются нераспознанными. Ваша задача — вычислить коэффициент распознавания. Коэффициент распознавания определяется как $R/A \cdot 100\%$, где R — количество распознанных символов и A — общее количество символов в тексте. При этом переводы строк символами не считаются.

Формат входного файла

Входной файл содержит как минимум одну непустую строку распознанного текста. Нераспознанные символы представлены символом "`#`". При этом длина строки не превосходит 100 символов, строки содержат только символы с ASCII-кодами от 32 до 126 включительно, при этом "`#`" не встречается в исходном тексте.

Формат выходного файла

Выведите коэффициент распознавания, округлённый до ближайшего числа, десятичная запись которого содержит не более одного знака после точки (0.05 округляется до 0.1). При этом лишние нули после точки выводить запрещено (то есть вывод 0.50 вместо 0.5 или 25.0 вместо 25 будет признан некорректным).

Пример

ocr.in	ocr.out
`Pr#nt the r#tio in perce##,` `rounde##to t#e nearest number`	87.7
`Professor Gromov.`	100

Задача N. Robots (Division 2 Only!)

Имя входного файла:	`robots.in`
Имя выходного файла:	`robots.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

Профессор Громов разработал концепцию роботов нового поколения. Сейчас он получил с завода пробную партию и изучает взаимодействие роботов.

Пронаблюдав за их взаимодействием, профессор предположил, что каждый робот определяет себя как объект некоторого пола (мужского или женского) и взаимоействует только с роботами противоположного пола.

Взаимодействие всех задействованных в эксперименте роботов запротоколировано. Реализовать это было просто, так как каждый робот помечен серийным номером. Вам дан список всех случаев взаимодействия между роботами. Выясните, подтверждает ли данный список версию Громова или опровергает её.

Формат входного файла

В первой строке входного файла заданы два целых числа: количество роботов N ($1 \le N \le 2000$) и количество взаимодействий K ($0 \le K \le 10^6$). В каждой из последующих K строк указаны серийные номера двух взаимодействующих роботов — целые числа от 1 до N (серийные номера роботов идут подряд, начиная с единицы, и для любых двух различных роботов различны).

Формат выходного файла

В случае, если эксперимент не противоречит предположению профессора, выведите "`OK`", в противном случае выведите "`Epic Fail`".

Пример

robots.in	robots.out
3 3 2 1 2 3 3 1	Epic Fail
5 2 2 1 3 4	OK

Задача O. Segments (Division 2 Only!)

Имя входного файла:	`segments.in`
Имя выходного файла:	`segments.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

На плоскости задано множество отрезков. Найти количество различных пар отрезков, которые перекрываются. Отрезки считаются перекрывающимися, если они имеют бесконечное количество общих точек.

Формат входного файла

В первой строке входного файла содержится целое число n — количество отрезков ($1 \le n \le 10^5$). Далее следуют n строк, каждая из которых содержит четыре целых числа x_1, y_1, x_2, y_2 ($0 \le x_1, y_1, x_2, y_2 \le 10^6$, $(x_1 - x_2)^2 + (y_1 - y_2)^2 > 0$), задающих отрезок, соединяющий точки (x_1, y_1) и (x_2, y_2).

Формат выходного файла

Выведите единственную строку — количество различных пар перекрывающихся отрезков в заданном множестве.

Пример

segments.in	segments.out
8 1 1 4 4 3 3 6 6 2 6 6 3 20 0 42 0 40 0 61 0 30 0 53 0 123 0 201 0 143 0 161 0	5
1 1 2 3 4	0

Задача P. Spinity (Division 2 Only!)

Имя входного файла:	spinity.in
Имя выходного файла:	spinity.out
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 Mebibytes

Некоторые строчные латинские буквы обладают следующим свойством: при повороте на 180 градусов они переходят или сами в себя, или в другие буквы. Можно составить даже целые слова, которые будут переходить сами в себя при повороте на 180 градусов, например, “sos” или “mow”. Введём для слова w параметр “вращаемость” $Sp(w)$ следующим образом:

Будем считать, что буквы ‘o’, ‘l’, ‘s’ ‘x’ автосимметричны при повороте, и что существуют следующие пары взаимно симметричных букв: ‘b’ и ‘q’, ‘d’ и ‘p’, ‘m’ и ‘w’, ‘n’ и ‘u’.

Для двух букв c_1 и c_2 $s(c_1, c_2)$ равно 3, если буквы совпадают и являются автосимметричными, 2, если буквы не совпадают, но составляют симметричную пару, 1, если буквы совпадают, но не являются автосимметричными, и 0 в оставшихся случаях.

Наконец, определим вращаемость $Sp(w)$ как сумму значений функции s для симметрично расположенных пар букв: для n-буквенного слова w $Sp(w) = s(w_1, w_n) + s(w_2, w_{n-1}) + \ldots + s(w_n, w_1)$.

Ваша задача — для каждого из заданных слов вычислить его вращаемость.

Формат входного файла

Входной файл состоит из не более, чем 100, и не менее, чем одного слова. Каждое слово расположено в отдельной строке и состоит из не менее, чем 1, но не более, чем 10 строчных латинских букв.

Формат выходного файла

Для каждого слова из входного файла выведите в отдельной строке само это слово и его вращаемость в соответствии с форматом, указанном в примере. Результаты выводить в том порядке, в котором слова следуют во входном файле.

Пример

spinity.in	spinity.out
spellers ololo meow axe pad sos nu	spellers 14 ololo 15 meow 4 axe 3 pad 5 sos 9 nu 4