

Для всех задач:

Имя входного файла: *input.txt*
Имя выходного файла: *output.txt*
Ограничение по памяти: *256 Мб*

Задача 1. Банкомат

Ограничение по времени на 1 тест: **2 сек.**
для задач на Java: **3 сек.**

Один отдаленный от центрального офиса банкомат пользовался большой популярностью у клиентов за возможность получать нужную сумму с точностью до рубля. Со временем некоторые кнопки в этом банкомате перестали нажиматься.

Как результат, некоторые граждане вообще не могут использовать на нем свою карту, так как не имеют возможности ввести пин-код. Другим же пользователям для получения денег приходится набирать сумму, нажимая только на работающие кнопки с цифрами.

После каждой выдачи денег карточка из банкомата извлекается. Поэтому для снятия желаемой суммы денег нужно запастись терпением и несколько раз вставлять карту в банкомат.

Определите, за какое минимальное количество использований карты можно снять всю нужную сумму.

Входные данные

В первой строке входного файла через пробел записано K цифр, соответствующих кнопкам, которые не работают ($6 \leq K \leq 10$).

Во второй строке записаны два целых числа P и S , P — это пин-код, состоящий из четырех цифр, а S — сумма, которую нужно снять с карточки ($1 \leq S < 10^6$).

Выходные данные

В выходной файл необходимо вывести целое число — минимальное количество использований карточки в данном банкомате для получения указанной суммы, после которого через пробелы выдать последовательность снимаемых сумм денег. Если таких последовательностей несколько, то вывести любую из них.

Если нужную сумму снять невозможно, то вывести **NO**.

Примеры

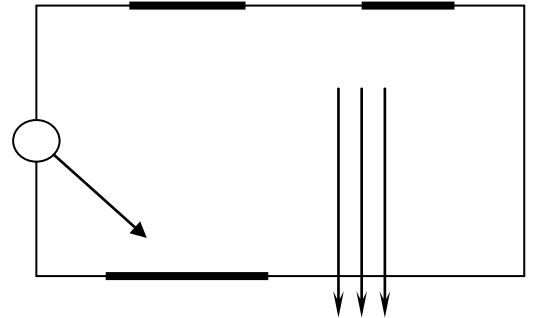
<i>input.txt</i>	<i>output.txt</i>
1 2 3 4 6 5 9889 78	1 78
1 2 3 4 5 6 9988 12	NO

Задача 2. Шарик на столе

Ограничение по времени на 1 тест: **1 сек.**

для задач на Java: **3 сек.**

Давным-давно в далёкой галактике металлический шарик катался по странному столу. Стол представляет собой прямоугольник, левый и правый края которого открыты, а верхний и нижний (в соответствии с видом сверху, как показано на рисунке) его края могут содержать участки с бордюрами. Стол расположен в зоне действия сильного поля, которое действует в вертикальном направлении. На рисунке направление поля показано тремя параллельными стрелками. Поле придаёт шарiku постоянную вертикальную скорость в направлении своего действия. Шарик стартует с середины левого края стола под углом 45° относительно его нижней стороны и движется с постоянной горизонтальной скоростью. Если шарик выкатывается за пределы стола, то он сваливается с него. В этом случае он вынужден начинать свой путь сначала. Удар шарика о бордюр является абсолютно неупругим: при ударе шарик теряет свою вертикальную составляющую скорости, но его горизонтальное движение продолжается вдоль бордюра. Если при этом поле не изменит своего направления, то по достижении конца бордюра шарик падает со стола.



Возникает заинтересованная сторона, которая желает довести шарик до другого края стола с помощью ручных переключений поля. Вручную поле переключается мгновенно. При этом, если шарик находится в свободном движении или катится по бордюру, по направлению к которому было включено поле, то после переключения поля шарик приобретает вертикальную компоненту скорости, равную горизонтальной, и направленную к противоположной стороне стола.

Требуется придумать эффективную схему переключений поля, которая позволит минимальным количеством ручных переключений довести шарик из начальной точки на левом крае стола до правого края, не позволяя ему свалиться на своем пути.

Входные данные

В первой строке входного файла через пробел записано два целых числа L и W — длина и ширина стола ($1 \leq L \leq 10^8$, $1 \leq W \leq 10^9$).

Во второй строке записано два целых числа N и M — количество бордюров на верхней и нижней сторонах стола соответственно ($0 \leq N, M \leq 10^5$).

В следующих N строках описываются бордюры, расположенные на верхней стороне стола. Каждый бордюр описывается на отдельной строке двумя целыми числами l_i и r_i , которые задают горизонтальные координаты начала и конца бордюра соответственно ($0 \leq l_i < r_i \leq L$).

В следующих M строках описываются бордюры, расположенные на нижней стороне стола в таком же формате.

Никакие два бордюра не пересекаются и не касаются. Горизонтальная координата, с которой стартует шарик, равна 0. Касание шарика бордюра в крайней его точке также считается ударом. Изначально поле направлено в сторону нижней границы стола.

Выходные данные

В первую строку выходного файла необходимо вывести единственное целое число — минимальное количество переключений поля, необходимое для того, чтобы довести шарик до правого конца стола.

Пример

<i>input.txt</i>	<i>output.txt</i>
<pre> 9 4 1 1 6 8 2 3 </pre>	<pre> 2 </pre>

Примечание: рисунок в условии не соответствует приведенному примеру.

Задача 3. Возведение Старгородского трамвая

Ограничение по времени на 1 тест: **3 сек.**
для задач на Java: **5 сек.**

Виктору Михайловичу Полесову после закупки партии плашек в три восьмых дюйма пришла следующая партия — в две седьмых дюйма. Он и призадумался — три восьмых да две седьмых это же будет тридцать семь пятьдесят шестых... А если следом придут плашки со знаменателем дроби девять, то оперировать уже придется с кучей дробей с единым знаменателем пятьсот четыре и так далее. Тут-то перед ним и стала во всём своем ужасе картина его грядущей деятельности по возведению Старгородского трамвая. А именно: каждую неделю ему придется решать задачу — какой комплект плашек (с каким знаменателем) из предложенных закупить для продолжения строительства.

Цель же гусара-одиночки с мотором — получить в итоге своей деятельности как можно больший наименьший общий знаменатель для всех партий плашек.

Входные данные

В первой строке входного файла записано целое число N — количество недель, на протяжении которых Виктор Михайлович выбирает знаменатели в партиях плашек ($1 \leq N \leq 20$).

В следующих N строках дано описание каждой из N недель поставок плашек.

Первым в каждой строке записано целое число k_i , задающее количество комплектов, предложенных на i -ой неделе ($1 \leq k_i \leq 20$), после него через пробел задаются k_i целых чисел a_{ij} — соответственно знаменатели в j -ой партии на i -ой неделе ($1 \leq a_{ij} \leq 42$).

Выходные данные

В выходной файл необходимо вывести одно число — максимально возможный наименьший общий знаменатель всех N партий, который может получиться, если выбирать каждую неделю по одному знаменателю из предложенных.

Пример

<i>input.txt</i>	<i>output.txt</i>
3 2 4 6 3 1 5 8 2 7 10	210

Задача 4. RPG

Ограничение по времени на 1 тест: **1 сек.**
для задач на Java: **2 сек.**

В играх жанра RPG (role-playing game) игроки управляют персонажами, которые характеризуются несколькими атрибутами. В нашей простой игре у персонажа всего два атрибута: сила *Str* и ловкость *Dex*. Изначально персонаж наделен некоторыми способностями, поэтому его атрибуты имеют некоторое начальное значение. Игроки стремятся повысить атрибуты своих персонажей с помощью надевания на них вещей. Вещи могут повышать атрибуты на фиксированную величину или не изменять их. Например, можно надеть на персонажа шлем, который увеличит его силу, но никак не будет влиять на ловкость. Все вещи делятся на заданное количество типов: шлемы, доспехи, перчатки, сапоги и так далее. Причем на персонаже может быть надета максимум одна вещь каждого типа. Задача выбора набора надетых вещей осложняется тем, что вещи имеют требования к атрибутам персонажа. Если требуется надеть вещь определенного типа, то на персонаже не должно быть надето вещи данного типа, и значение каждого атрибута с учетом бонусов от остальных надетых на персонажа вещей не должно быть меньше, чем требование к этому атрибуту у заданной вещи.

Однажды вашему персонажу повезло, он нашел клад, в котором было большое количество вещей. Но унести вещи он сможет, только надев их на себя. Вам приглянулась одна вещь, которую вы хотите, во что бы то ни стало, унести с собой. Ваша задача определить, существует ли такой порядок надевания вещей, чтобы в итоге надеть ту самую вещь и уйти с трофеями.

До того как был найден клад, на персонаже не было ни одной вещи.

Входные данные

В первой строке входного файла записано два целых числа K и N — количество типов вещей и общее количество различных вещей ($1 \leq K \leq 5$, $1 \leq N \leq 3000$).

Во второй строке описаны базовые значения атрибутов персонажа: целые числа *Str* и *Dex* ($0 \leq Str, Dex \leq 100$).

Далее идут N строк, каждая из которых описывает одну вещь. Описание каждой вещи содержит пять целых чисел: $Type_i$, $RStr_i$, $RDex_i$, $AStr_i$, $ADex_i$ — соответственно тип вещи, ее требования к атрибутам и ее бонусы к атрибутам ($1 \leq Type_i \leq K$, $0 \leq RStr_i, RDex_i, AStr_i, ADex_i \leq 101$).

Считается, что требуется надеть первую вещь.

Выходные данные

В единственную строку выходного файла необходимо вывести слово **YES** или слово **NO**. **YES** должно быть выведено, если указанную вещь можно надеть, **NO** — в противном случае.

Примеры

<i>input.txt</i>	<i>output.txt</i>
1 1 10 20 1 0 10 10 10	YES
1 1 10 10 1 0 20 10 10	NO

Задача 5. Казино «Аккордеон»

Ограничение по времени на 1 тест: **1 сек.**

Мало кто знает, что роман «Казино «Рояль» Яна Флеминга написан по мотивам реальных событий. Тем более, почти никто не знает, где и как разворачивались эти действия. Но вам в руки попался обрывок шифровки, и ситуация оказалась очень интересной...

1943 год, Берлин. Престижное казино «Аккордеон», куда допускают только истинных арийцев. За покерным столом сидят Генрих Мюллер и Макс Отто фон Штирлиц и играют. Ставкой являются секретные документы. Конечно же, честный русский разведчик Максим Максимович Исаев не стал бы играть при таких крупных ставках, если бы не был абсолютно уверен в своей победе. Но, так как он уже сел за стол, то следует полагать, что у него есть пара тузов в рукаве. Мюллер тоже догадывался об этом и попросил Штирлица закатать рукава.

Благодаря хитроумной системе зеркал, Штирлицу удалось разглядеть карты Мюллера. Из пяти карт, лежащих на столе, перевернуто три карты. О содержимом двух остальных разведчик может только догадываться. Штирлиц может лишь оценить вероятность выигрыша и, соответственно, сделать ставку. Для более удобного расчёта в конструкторском бюро были разработаны специальные часы марки «Игельс».

Написать для них программу расчёта поручено вам.

Правила игры в покер:

Каждому игроку сдаются две карты втёмную. Затем в центр стола выкладывается прикуп из пяти карт в «светлую». Это общие для всех игроков карты. Каждый игрок может пользоваться прикупом для образования лучшей пятикарточной комбинации. Из двух карт игрока и пяти карт прикупа выбираются пять карт, составляющие лучшую комбинацию. Банк забирает игрок, оставшийся в игре к моменту открытия последней карты прикупа и имеющий лучшую пятикарточную комбинацию, составленную из своих карт и карт прикупа (в любой пропорции). Прикуп выкладывается в три этапа. Сначала выкладываются три карты, затем ещё одна карта, затем последняя карта.

Покерные комбинации:

Ниже представлены комбинации различных уровней, от самых слабых до самых сильных.

	Старшая карта	Когда не собрана ни одна из перечисленных ниже комбинаций.
	Одна Пара	Две карты одного достоинства.
	Две Пары	Любые две карты одного достоинства вместе с двумя картами другого достоинства.
	Тройка	Три карты одного достоинства.
	Стрит	Пять последовательных карт. Туз может быть взят как высокая или низкая комбинация, но не одновременно высокая и низкая в одной и той же руке.
	Флеш	Пять карт одной масти.

	Фулл-хаус	Три карты одного достоинства вместе с двумя картами другого достоинства.
	Каре	Четыре карты одного достоинства.
	Стрит-флеш	Стрит из пяти карт одной масти.
	Флеш-рояль	Стрит-флеш от 10 до туза.

Так как события происходят в военное время, то ведётся лишь подсчёт стоимости комбинации, стоимость карт в ней не учитывается. Так, например, если сильнейшей комбинацией Штирлица будет пара двоек, а Мюллера — пара тузов, будет объявлена ничья.

Входные данные

В первой строке входного файла находится описания двух карт Штирлица. Это две последовательности по два символа, где первый символ обозначает достоинство карты (2...9 — соответствующая цифра, 10 — Т, валет — J, дама — Q, король — K, туз — A), второй символ — масть (S — пики, C — крести, H — червы, D — бубны).

Во второй строке в таком же формате задано описание карт Мюллера.

В третьей строке — описания трех «общих» карт, лежащих на столе.

Выходные данные

В выходной файл необходимо вывести три вещественных числа: вероятность выигрыша, ничьей и проигрыша Штирлица соответственно. Значения должны представляться в виде десятичной дроби с точностью до 10^{-4} .

Примеры

<i>input.txt</i>	<i>output.txt</i>
2H 7S AD AC 2C 7C AH	0.00200 0.165657 0.832323
AC KC AH KH QC JC TC	1.000000 0.00000 0.00000

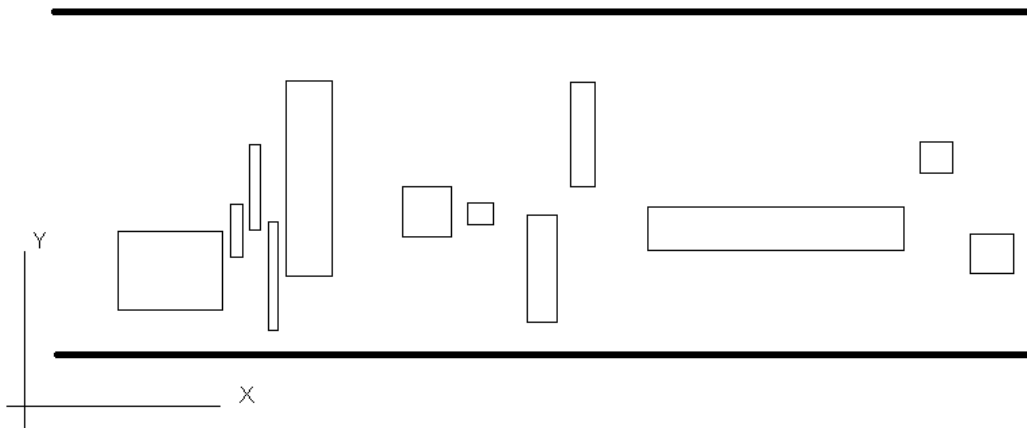
Задача 6. Ремонт дороги на Пирогова

Ограничение по времени на 1 тест: 1 сек.

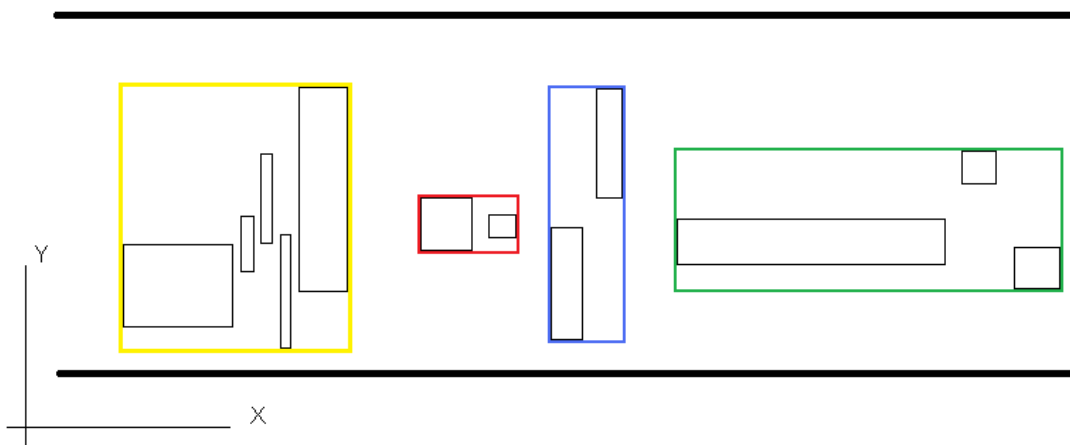
*Выбоина на выбоине и
выбоиной погоняет.*

До недавнего времени улица Пирогова была почти непроходима для транспорта. Водители легковых автомобилей старались объезжать ее стороной. Наконец, администрация решила сделать на дороге косметический ремонт — заделать все выбоины, причем, в кратчайшие сроки.

Составили план дороги. На плане дорога — прямоугольник, левая сторона которого совпадает с началом координат по оси ОХ. Отметили на плане все выбоины. Оказалось, что они имеют форму прямоугольников со сторонами, параллельными осям координат, при этом отрезки-проекции выбоин на ось ОХ не пересекаются.



Время T , которое уходит на заливку одного прямоугольника площадью S вычисляется по формуле: $T = t_0 + S / v$. Ремонтники умеют класть асфальт только прямоугольниками со сторонами, параллельными осям координат, и могут заливать не только одну выбоину отдельно, но и группами, за счет чего может сократиться время работы.



При этом, если большим прямоугольным куском асфальта заливаются с i -ой по j -ю выбоины, по порядку их следования вдоль оси ОХ, то должны полностью покрыться и все те, что между ними.

Нужно так сгруппировать выбоины, чтобы минимизировать время работы.

Входные данные

В первой строке входного файла записано через пробел три числа n , t_0 и v , где n — целое число, количество выбоин на дороге, t_0 и v — вещественные числа ($1 \leq n \leq 3000$, $0 \leq t_0 \leq 7000$, $0 < v \leq 1000$).

В следующих n строках даны координаты выбоин в порядке их встречаемости на плане вдоль оси ОХ. Каждая выбоина задается четырьмя вещественными числами — координатами левого нижнего и правого верхнего своих углов. Известно, что ширина дороги не превышает 2000, а длина не больше $3,5 \cdot 10^6$.

Гарантируется, что входные данные таковы, что ответ не превышает 10^8 .

Выходные данные

В выходной файл необходимо вывести одно вещественное число с точностью до 10^{-3} — минимальное время, требуемое для ремонта дороги.

Пример

<i>input.txt</i>	<i>output.txt</i>
3 7.2 1 1 1 2 3 3 3 4 5 5 2 6 3	25.4

Задача 7. Пароли

Ограничение по времени на 1 тест: 2 сек.

В корпорации Crazel (Crazy Electronics) исследуют возможность использования процессоров, работающих на системах счисления с различными основаниями. В разрабатываемом процессоре элементарной ячейкой памяти является крабайт (crabyte). Крабайт состоит из Q крабит (crabit). Каждый крабит может принимать значения от 0 до $P-1$, где P – простое число. Таким образом, один крабайт памяти может хранить целые числа в диапазоне от 0 до P^Q-1 включительно.

Арифметика на подобном процессоре выполняется по модулю P^Q . При сложении или умножении двух крабайтовых чисел в ячейку результата записывается лишь остаток от деления на P^Q .

Джордж — гений, работающий в корпорации Crazel. Как и все гении, он очень рассеян, и вообще человек не от мира сего. В частности, его мозг работает специфически. Например, он всегда вспоминает необходимые пароли, но обычно не с первой попытки. А на некоторых сайтах после трёх неудачных попыток ввода пароля аккаунт блокируется. Чтобы избежать подобных проблем, он решил сохранять хеши от паролей в памяти прототипа разрабатываемого процессора. Для этого он записал пароль в память процессора, а затем запустил собственную программу, которая вычисляет хеш пароля по следующей формуле:

$$Y_i = \sum_{j=1}^N A_{ij} X_j$$

X_j – содержимое j -го крабайта пароля,

Y_i – содержимое i -го крабайта результирующего хеша,

A_{ij} – записанный в крабайте элемент фиксированной матрицы.

Майкл хочет украсть один из паролей Джорджа. Он узнал хеш, записанный в памяти прототипа, а также матрицу, используемую программой Джорджа. Он просит вас помочь ему в осуществлении коварного плана.

Найдите количество паролей, имеющих заданный хеш, а также выведите любой из них.

Входные данные

В первой строке входного файла записано четыре целых числа: N – длина пароля в крабайтах, M – длина хеша в крабайтах ($1 \leq N, M \leq 200$), P – количество возможных состояний крабита ($2 \leq P < 10^9$) и Q – количество крабит в крабайте ($1 \leq Q < 30$). В следующей строке записано M чисел – содержимое крабайтов хеша. Затем идёт M строк по N чисел в каждой – матрица, заложенная в программу Джорджа.

Гарантируется, что P – простое число. Чтобы было проще эмулировать работу прототипа на обычном 32-битном процессоре, количество возможных состояний крабайта не превосходит 10^9 .

Выходные данные

Первая строка выходного файла должна содержать информацию о количестве паролей, имеющих заданный хеш. Если таких паролей нет, выведите слово **Fool**. Если паролей менее миллиарда, то выведите слово **Little** и количество паролей через пробел. Если паролей не меньше миллиарда, то выведите **Many** и остаток от деления количества паролей на $(10^9 + 7)$ через пробел.

В случае, когда искомые пароли существуют, выведите во вторую строку N чисел через пробел – содержимое крабайт любого из этих паролей.

Примеры

<i>input.txt</i>	<i>output.txt</i>
5 2 7 1 3 3 1 1 1 1 1 2 2 2 2 2	Fool
2 3 5 2 3 6 12 1 1 2 2 4 4	Little 25 3 0
5 5 2 10 0	Many 898961331 2 3 5 7 11

Задача 8. Памятники

Ограничение по времени на 1 тест: **2 сек.**
для задач на Java: **4 сек.**

В связи с 20-летием со дня распада СССР в городе Н-ске решили поставить два памятника. Один памятник должен изображать серп, а другой, соответственно, молот. Руководить изготовлением памятников поручили Департаменту культурного наследия (ДКН) города Н-ска.

Каждый памятник планируется сделать из гранита, причём две болванки гранита уже лежат на складе. Остаётся лишь обработать болванки, чтобы получить готовые памятники. В городе Н-ске имеются цеха, которые могут принять участие в обработке памятников. Поскольку каждый цех хочет поучаствовать в проекте, но не каждый цех может выполнять все необходимые функции, то решено было разделить работу на части.

ДКН составил планы обработки памятников. В каждом плане перечислены все части работы в порядке их выполнения. Для каждой части имеется подробная техническая документация, указаны цех, который должен эту часть работы выполнить, и расчетное время, необходимое для выполнения. Никакой цех не может вести работу над обоими памятниками одновременно, а значит, может возникнуть ситуация, когда один из памятников придётся временно поместить обратно на склад. Цех может прервать обработку памятника ночью и переключиться на работу над другим памятником. В этом случае памятник отправляется на склад до возобновления работы над ним.

Перед ДКН поставили задачу закончить оба памятника к 26 декабря 2011 года (день X). Если памятник будет закончен на A дней позже, то из ДКН уволят A^2 служащих. Вам поручили следить за ходом выполнения работы. Поскольку вы не хотите остаться без работы, вам нужно избежать увольнений, если это возможно, или минимизировать суммарное количество увольнений в противном случае.

Входные данные

Во входном файле сначала описан план работы над памятником «Серп», а затем план работы над памятником «Молот».

Описание плана работы начинается с целого числа K – количества частей, на которые разделена работа над памятником ($1 \leq K \leq 400$). В следующих K строках содержится информация о частях в порядке их выполнения. Для каждой части записано название цеха, который будет выполнять эту часть работы, а также необходимое для этого время в днях – целое число от 1 до 10^6 включительно. Название имеет длину от 1 до 1000 символов и состоит из латинских букв и цифр. Техническая документация для решения задачи вам не потребуется. Каждый цех имеет своё уникальное имя.

В последней строке записано целое число T – количество рабочих дней, которое осталось до дня X ($0 \leq T \leq 10^9$).

Выходные данные

В первую строку выходного файла выведите минимально возможное количество увольнений. Затем выведите подробную историю ведения работ над памятником «Серп» и аналогичную историю для памятника «Молот». Минимальное количество увольнений должно достигаться при осуществлении этих историй.

История ведения работ над памятником начинается с целого числа P – количества отрезков времени в истории ($1 \leq P \leq 10000$). Каждая из последующих P строк должна содержать запись об отрезке времени. Запись состоит из целых чисел B , E и строки S , записанных через пробел. B – номер первого дня отрезка, E – номер последнего дня отрезка, S – название цеха, в котором деталь находится в этом отрезке времени ($1 \leq B \leq E \leq 10^9$). Если деталь находится на складе, то S равно “[storehouse]” (без кавычек). Первый день работы имеет номер 1. Отрезки времени не должны перекрываться, они должны идти подряд без пропусков дней, первый отрезок должен начинаться с первого дня работы.

Примеры

<i>input.txt</i>	<i>output.txt</i>
2 Zubilo 3 Nazhdachka 2 2 Kualda 2 Kley 1 10	0 2 1 3 Zubilo 4 5 Nazhdachka 4 1 1 Kualda 2 2 Kualda 3 3 [storehouse] 4 4 Kley
4 Floor1 3 Floor2 1 Floor4 2 Floor4 3 3 Floor5 2 Floor2 3 Floor4 2 7	17 5 1 3 Floor1 4 4 Floor2 5 6 Floor4 7 8 [storehouse] 9 11 Floor4 5 1 2 Floor5 3 3 Floor2 4 4 [storehouse] 5 6 Floor2 7 8 Floor4

Задача 9. Модная шляпка

Ограничение по времени на 1 тест: **1 сек.**

Незнайка на день рождения Кнопочке решил подарить модную шляпу. Он раздобыл каталог шляп и выбрал фасон. К сожалению, в магазинах, которые он знал, продавались только конические шляпы, как у него. Он решил переделать коническую шляпу в модный вариант. Для этого какую-то часть шляпы нужно было отрезать. Но он не знал, как разрезать шляпу, чтобы получить нужный фасон. Поэтому он пошел к Знайке за советом. Знайка с легкостью свел эту задачу к математической.

Шляпа — это конус, заданный уравнением $x^2 + y^2 = z^2$, который нужно разрезать плоскостью, описываемой уравнением $Ax + By + Cz = D$.

Осталось только определить, какой тип кривой второго порядка получится при разрезе конуса этой плоскостью с заданными коэффициентами уравнения.

Входные данные

В единственной строке входного файла записано четыре вещественных числа **A**, **B**, **C** и **D**. Все числа по модулю не превосходят 1000 и имеют не более чем шесть знаков после запятой. Гарантируется, что из первых трех чисел, хотя бы одно — ненулевое.

Выходные данные

В единственной строке выходного файла должно быть записано одно слово, в зависимости от полученного типа кривой:

circle — окружность

ellipse — эллипс, кроме окружности

empty — пустое множество (мнимый эллипс)

hyperbola — гипербола

intersecting — пара пересекающихся прямых

parabola — парабола

parallel — пара несовпадающих параллельных прямых

point — точка

straight — прямая (пара совпадающих прямых)

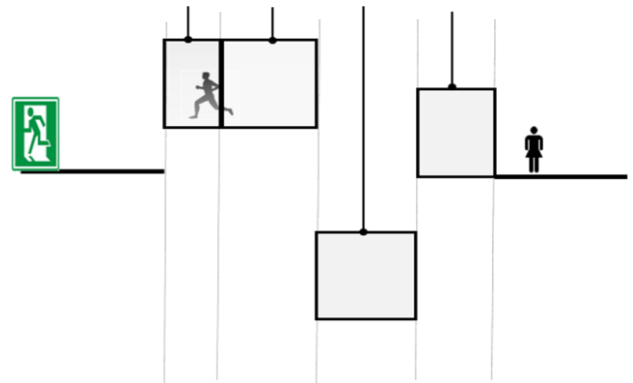
Пример

<i>input.txt</i>	<i>output.txt</i>
1.0 0.0 2.0 3.0	ellipse

Задача 10. Побег

Ограничение по времени на 1 тест: **1 сек.**

Остап Бендер убегает от разъяренной мадам Грицацуевой через длинный коридор, разделенный сложной системой лифтов, располагающихся последовательно один за другим и непрерывно едущих вверх-вниз. Оба конца коридора расположены на одном уровне. При этом и товарищ Бендер, и вдова могут входить в лифт, выходить из него, а также пересаживаться с лифта на лифт при условии, что лифты находятся на одинаковой высоте друг с другом или с коридором. Добежав до конца коридора, он запирает спасительную дверь. После этого бриллиантовой вдовушке остается только лишь поудобней устроиться на ночлег.



Наша цель — помочь Остапу найти минимальную скорость, достаточную для того, чтобы брошенная вдова не смогла его догнать.

Входные данные

В первой строке входного файла приведено пять положительных целых чисел t, v, a, N, b : t — время старта вдовы (Остап стартует в нулевой момент времени), v — скорость мадам Грицацуевой, a — длина первоначального отрезка коридора до первого лифта, N — количество лифтов, b — длина последнего участка коридора (расстояние от последнего лифта до спасительной двери). Все эти числа не превосходят 1000.

В следующих N строках приводится описание каждого из N лифтов — по шесть целых чисел в строке. Описание i -го лифта задается числами V_i, T_i, H_i, TO_i, W_i и Z_i .

V_i — скорость вертикального перемещения лифта, она постоянна, меняется только ее направление — вверх или вниз ($1 \leq V_i \leq 1000$).

T_i — полупериод поездки лифта — время, которое он будет ехать вверх или вниз ($1 \leq T_i \leq 1000$).

H_i — первоначальная высота положения лифта, отсчитываемая от уровня коридора, который принимается за 0 ($-1000 \leq H_i \leq 1000$).

TO_i — время, отсчитываемое от начального момента времени до первой смены направления ($1 \leq TO_i \leq T_i$).

W_i — длина платформы лифта (расстояние, которое необходимо преодолеть, чтобы пересечь на следующий лифт, $1 \leq W_i \leq 1000$).

Z_i принимает одно из двух значений — либо **1**, либо **-1**. Оно показывает первоначальное направление движения лифта — вверх или вниз соответственно.

Выходные данные

В выходной файл необходимо вывести одно вещественное число, округленное до трех знаков после запятой, минимальную скорость, достаточную для того, чтобы убежать от мадам Грицацуевой. Гарантируется, что эта скорость не превосходит 10^6 . Если Остапу убежать никак не удастся, нужно вывести **-1**.

Примеры

<i>input.txt</i>	<i>output.txt</i>
2 1 2 1 3 2 2 2 1 4 1	1
1 1 2 1 3 2 2 2 1 4 1	-1