Problem A. Лотерея

Input file:	stdin
Output file:	stdout
Time limit:	1 second
Memory limit:	256 megabytes

В последние годы телевизионные лотереи стремительно теряют свои рейтинги, уступая значительную часть рынка интернет-казино и букмекерским конторам. Чтобы как-то исправить сложившуюся ситуацию и поднять степень доверия населения, организаторы одной из лотерей внедрили новую честную и прозрачную схему определения победителя.

Все лотерейные билеты пронумерованы последовательно числами от L до R (включительно). На основе исходных номеров билетов и некоторого случайного числа X формируется последовательность зашифрованных номеров A, i-й ($1 \le i \le R - L + 1$) элемент которой равен $(L + i - 1) \oplus X$, где $\oplus$ означает операцию поразрядного сложения двоичных представлений чисел по модулю 2.

Специально приглашенная на шоу звезда, которой неизвестно значение числа X, выбирает некоторое число K ($1 \le K \le R - L + 1$), после чего выигрышным объявляется билет, зашифрованный номер которого является K-м элементом последовательности A в порядке возрастания.

Input

В единственной строке заданы числа L, R, X и K ($0 \le L \le R \le 10^{18}$, $0 \le X \le 10^{18}$, $1 \le K \le R - L + 1$).

Output

Исходный номер выигрышного билета.

Examples

stdin	stdout
6 15 0 7	12
1 20 10 5	14

Problem B. Турникмены

Input file:	stdin
Output file:	stdout
Time limit:	1 second
Memory limit:	256 megabytes

Турникменами относительно недавно стали называть себя уличные спортсмены, выполняющие силовые упражнения и трюки на обычных дворовых турниках. Конечно, и раньше многие молодые люди тренировались на турниках, выполняя сложные элементы, но лишь с появлением социальных сетей и видеохостингов движение турникменов приобрело глобальные масштабы.

Но с чего начать тем, кто подтягивается всего пару-тройку раз за один подход? Ответ прост и известен давно — с «лесенки». Так называют коллективную игру, участники которой по очереди совершают подходы, выполняя одинаковое число подтягиваний. С каждым новым подходом число подтягиваний увеличивается, пока не достигнет некоторого максимального значения, после чего начинает уменьшаться. Игра оканчивается, когда необходимое число подтягиваний в подходе становится равным нулю.

Чтобы тренировки были по-настоящему эффективными, важно правильно выбрать максимальное число подтягиваний, выполняемых за один подход в игре. Рассмотрим один из вариантов игры, позволяющий двум участникам тренироваться вместе, даже если они обладают разной квалификацией.

Пусть квалификация первого игрока равна A, а квалификация второго равна B. Обозначим количество подтягиваний в подходе для первого игрока числом N, а для второго — числом M. В первом подходе, в качестве обязательной разминки, оба игрока выполняют по одному подтягиванию (т.е. изначально $N = M = 1$). После выполнения первого подхода обоими игроками вступает в силу правило: если $N \times A < M \times B$, то число N увеличивается на единицу и первый игрок выполняет подход, а если $N \times A > M \times B$, то число M увеличивается на единицу и второй игрок выполняет подход. Исключительными являются ситуации, в которых либо один из игроков должен выполнить второй подход подряд (без учета самого первого подхода), либо после завершения некоторого подхода выполняется условие $N \times A = M \times B$, — в этих случаях правило, описанное выше, перестает действовать, и каждый игрок просто повторяет все свои подходы, выполняя их в обратном порядке, после чего игра завершается.

Ваша задача — зная квалификацию игроков, определить общее число подтягиваний, которое совершат оба игрока за время игры.

Input

В первой строке заданы два целых числа A и B ($1 \le A, B \le 10^9$) — квалификация первого и второго игрока, соответственно.

Output

Общее число подтягиваний, совершаемых за время игры обоими игроками.

Examples

stdin	stdout
3 5	18
1 2	8

Problem C. Маршрутный беспорядок

Input file:	stdin
Output file:	stdout
Time limit:	1 second
Memory limit:	256 megabytes

Маршрутные такси, появившиеся в огромных количествах на улицах Российских городов в 90-е годы, не только стали новым быстрым и удобным средством передвижения, но и привнесли некоторый беспорядок в городские транспортные системы. Многие водители регулярно отклонялись от своих маршрутов, пытаясь объехать пробки и набрать больше пассажиров, а некоторые маршрутные такси вообще перемещались по городу без каких-либо опознавательных знаков. Такое положение дел не устраивало городскую администрацию, поэтому было решено ужесточить контроль соблюдения маршрутов и ввести для них новую нумерацию.

В городе есть N перекрестков, пронумерованных от 1 до N, а также M дорог с двусторонним движением, соединяющих некоторые пары перекрестков. Между каждой парой перекрестков существует хотя бы один путь.

Решено организовать R маршрутов. Пусть для некоторого маршрута задана последовательность опорных перекрестков $A_1, A_2, \ldots, A_k$. Маршрутное такси должно начать движение от перекрестка A_1, после чего проехать по кратчайшему пути (по числу проезжаемых перекрестков) до A_2, от A_2 аналогичным образом до A_3, от A_i до A_{i+1} (для $1 \le i \le k-1$), и завершить движение в перекрестке A_k. При этом, если между некоторой парой перекрестков A_i и A_{i+1} есть несколько кратчайших путей, то среди них выбирается лексикографически наименьший.

Рассмотрим два различных пути X и Y с одинаковым числом вершин k. Путь X считается лексикографически меньше пути Y, если существует такое i ($1 \le i \le k$), что $X_i < Y_i$, и для всех j ($1 \le j < i$) верно, что $X_j = Y_j$.

Если все перекрестки, в которых побывает маршрутное такси на пути следования от A_1 до A_k, записать в последовательность B_1, B_2, ..., B_s, то номером маршрута будет сумма $\sum_{i=1}^{s} iB_i$. Если у нескольких маршрутов получаются одинаковые номера, то ко всем номерам, кроме первого, добавляют специальный код. Например, если получилось 4 маршрута с номером 8172, то у первого маршрута останется номер 8172, у второго 8172#2, у третьего 8172#3, у четвертого 8172#4.

Ваша задача — вычислить номера всех маршрутов, согласно описанной схеме.

Input

В первой строке заданы два числа N ($2 \le N \le 10^3$) и M ($1 \le M \le 10^4$).

Следующие M строк содержат информацию о дорогах. Каждая дорога описывается в отдельной строке двумя числами u и v ($1 \le u, v \le N$, $u \ne v$). Между каждой парой перекрестков не более одной дороги. На всех дорогах двустороннее движение.

В следующей строке задано число R ($1 \le R \le 10^5$) — количество маршрутов.

Далее в R строках задается описание маршрутов, по одному на каждую строку. Описание i-го маршрута содержит число k_i ($2 \le k_i \le N$) — число опорных перекрестков i-го маршрута и k_i чисел $A_{i,j}$ ($1 \le A_{i,j} \le N$) — номера опорных перекрестков i-го маршрута в порядке следования. Опорные перекрестки каждого отдельного маршрута попарно различны.

Сумма всех k_i не превышает $2 \cdot 10^5$.

Output

Выведите всего R строк, по одной на каждый маршрут. В i-й строке выведите номер i-го маршрута.

Examples

stdin	stdout
13 17 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 5 5 7 1 11 11 12 12 2 11 13 13 10 10 9 9 4 11 10 4 5 1 3 13 12 7 8 1 2 3 11 13 12 4 7 3 1 7 2 2 6 8	644 644#2 211 40

Problem D. Вахтер

Input file:	stdin
Output file:	stdout
Time limit:	1 second
Memory limit:	256 megabytes

Работать вахтером в студенческом общежитии — это настоящие испытание для нервов любого человека. Целый день студенты носятся мимо вахтера, пререкаются, когда их просят показать пропуск, и самое главное — постоянно хлопают дверью!

Одному из вахтеров надоело тратить свои нервы почем зря, и он решил по возможности сдерживать свой гнев. В течении дня он пообещал себе не реагировать на студентов до тех пор, пока суммарная громкость дверных хлопков S не станет большей либо равной пределу его терпения L. Как только это происходит, вахтер догоняет хлопнувшего дверью студента и читает ему длинную нотацию. После этого вахтер вновь успокаивается, а суммарная громкость S — обнуляется.

По известной громкости A_i каждого дверного хлопка за день, Вам необходимо подсчитать число прочитанных вахтером нотаций.

Input

В первой строке заданы два числа N ($1 \le N \le 1000$) и L ($1 \le L \le 1000$) — количество дверных хлопков и предел терпения вахтера. Во второй строке задано N чисел A_i ($1 \le A_i \le 1000$) — громкости дверных хлопков.

Output

Число прочитанных за день нотаций.

Examples

stdin	stdout
6 3 1 3 2 2 3 2	3

Problem E. Нумизматы

Input file:	stdin
Output file:	stdout
Time limit:	1 second
Memory limit:	256 megabytes

Нумизматами называют людей, коллекционирующих монеты. Многие делают это для души, пытаясь почувствовать дух ушедших времен, для других же монеты — это в первую очередь способ заработать.

Большинство нумизматов коллекционируют монеты, принадлежащие только определенной эпохе. Чтобы покупателям было легче ориентироваться при выборе монет, продавцы вынуждены постоянно поддерживать правильный порядок монет на витрине, упорядочивая их по возрастанию даты чеканки. В процессе демонстрации и покупки товара этот порядок часто нарушается, и для решения этой проблемы был разработан специальный алгоритм сортировки монет.

Пусть N — количество монет на витрине. За одну операцию продавец может выбрать некоторую группу стоящих подряд монет на витрине, длина которой обязательно должна равняться некоторой степени двойки $(2, 4, 8, \dots)$, и переместить самую старую монету группы в её начало, передвинув на одну позицию от начала все монеты группы, следовавшие ранее до самой старой.

Необходимо для заданного количества монет подсчитать минимальное число операций, которое потребуется для их упорядочивания по возрастанию даты чеканки согласно описанному алгоритму в худшем случае.

Input

В первой строке задано количество монет на витрине N $(1 \le N \le 10^9)$.

Output

Искомое число операций.

Examples

stdin	stdout
3	3

Problem F. Трамвайщики

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Трамвайщиками называют людей, для которых трамвай — это не просто экологически чистый вид транспорта, позволяющий избежать пробок, но и самый настоящий объект обожания, а также главная достопримечательность любого города. Именно эти люди организуют в городах комитеты и митинги в защиту трамваев, которые местные власти пытаются закрывать, якобы вследствие их нерентабельности. Свой отпуск трамвайщики часто посвящают путешествиям по стране и миру, в надежде увидеть и запечатлеть на фотографии редкие модели трамваев.

В один из городов, в котором по слухам есть несколько редких моделей трамваев, прибыла группа трамвайщиков. Раздобыв схему трамвайных маршрутов, они разделились и отправились на поиски. Когда один из трамвайщиков видит редкую модель трамвая, он тут же начинает обзванивать своих друзей, сообщая им конечные пункты маршрута его следования.

Трамвайная сеть города представляет собой связное множество из N остановок и $N-1$ участков путей между ними. Каждый трамвайный маршрут представляют собой кратчайший путь между некоторыми двумя остановками. Ваша задача — зная конечные остановки A и B редкого трамвая, определить минимальное число остановок, которое необходимо проехать трамвайщику, находящемуся на остановке C, чтобы оказаться на маршруте его следования.

Input

В первой строке задано число остановок в трамвайной сети N ($1 \le N \le 10^5$). В следующих $N-1$ строках задаются пары чисел U_i и V_i ($1 \le U_i, V_i \le N$) — номера остановок, которые соединяет i-й участок пути.

В следущей строке задано число запросов M ($1 \le M \le 10^5$). Следующие M строк содержат описания запросов в виде троек чисел A_i, B_i и C_i ($1 \le A_i, B_i, C_i \le N$), где A_i и B_i — конечные остановки редкого трамвая, а C_i — остановка, на которой находится трамвайщик.

Output

Выведите M строк, содержащих ответы на запросы, по одному в каждой. Ответом на i-й запрос является минимальное число остановок на пути трамвайщика от остановки C_i до маршрута следования трамвая с конечными остановками A_i и B_i.

Examples

stdin	stdout
6 1 2 6 2 3 1 3 4 3 5 3 1 2 3 6 4 5 4 3 2	1 1 2

Problem G. Входной контроль

Input file:	stdin
Output file:	stdout
Time limit:	1 second
Memory limit:	256 megabytes

На любом крупном мероприятии, будь то концерт или футбольный матч, в целях безопасности у входа на территорию арены устанавливаются турникеты. Люди выстраиваются в очереди к турникетам в ожидании разрешения на проход, а сотрудники полиции контролируют соблюдение порядка при заполнении стадиона.

У входа на стадион в ряд установлено N турникетов. С определенным интервалом на территорию арены пропускают несколько человек от некоторой группы стоящих подряд турникетов (по одному человеку от каждого турникета). Группа из k турникетов считается хорошо контролируемой, если для неё существует такое число $p \geq 1$, что в каждом из первых p турникетов зрители имеют попарно различные номера трибун на билетах ($A_i \neq A_j$ для всех $i \neq j, 1 \leq i, j \leq p$), а для оставшихся ($p < i \leq k$) верно: $A_i = A_{i-p}$.

Для каждого турникета известен номер трибуны в билете первого зрителя, стоящего в очереди к нему. Вам необходимо научиться определять, является ли некоторая группа стоящих подряд турникетов в данный момент хорошо контролируемой или нет.

Input

В первой строке задано число турникетов N ($1 \leq N \leq 10^5$). Во второй строке содержатся N целых чисел A_i ($1 \leq A_i \leq 10^5$) — номера трибун в билетах у зрителей, по одному для каждого турникета.

В третьей строке задано число запросов M ($1 \leq M \leq 10^5$). Следующие M строк содержат описания запросов, по одному в каждой строке. Запрос задается двумя числами L_i и R_i ($1 \leq L_i \leq R_i \leq N$) — границами индексов турникетов, входящих в i-ю проверяемую группу.

Output

Выведите строку S из M символов. Если группа турникетов, заданная в запросе i, хорошо контролируемая, то i-й символ строки S должен быть равен 1, иначе 0.

Examples

stdin	stdout
10 7 1 2 3 1 2 3 3 1 7 7 1 10 1 4 1 5 2 6 2 7 2 8 8 10	0101101
5 1 1 1 2 1 4 1 3 1 4 1 5 3 4	1001

Problem H. Игра на репетиции

Input file:	stdin
Output file:	stdout
Time limit:	2 seconds
Memory limit:	256 megabytes

Во время репетиций рок-группы, которые проводятся в гараже, у гитариста постоянно расстраивались струны. Двое других участников группы — басист и ударник — случайно нашли в гараже шахматную фигуру коня и, чтобы как-то скоротать время, придумали новую игру.

Игра ведется на бесконечной шахматной доске, причем координаты клеток могут быть как положительными, так и отрицательными. Перед началом игры фигура коня помещается в клетку с координатами $(X_0; Y_0)$. Двое игроков ходят по очереди. За один ход игрок должен переместить фигуру коня в одну из 8 клеток в соответствии с обычными правилами шахмат, но при этом для i-го хода должно выполняться условие $(|X_i| + |Y_i| < |X_{i-1}| + |Y_{i-1}|)$. Если игрок не может сделать ход, он объявляется проигравшим.

Ваша задача - определить победителя и суммарное число ходов при условии, что оба игрока действуют оптимально. При этом известно, что победитель пытается растянуть игру как можно дольше, а проигравший, напротив, хочет завершить как можно быстрее.

Input

В первой строке заданы координаты фигуры коня перед началом игры X_0 и Y_0 $(|X_0|+|Y_0| \leq 10000)$.

Output

В первой строке в случае победы первого игрока выведите «WIN», в случае победы второго игрока — «LOSE». Кавычки выводить не нужно.

Во второй строке выведите суммарное число ходов, сделанных обоими игроками.

Examples

stdin	stdout
5 5	WIN 3
-8 3	LOSE 4

Problem I. Заказное тестирование

Input file:	stdin
Output file:	stdout
Time limit:	1 секунда
Memory limit:	256 мегабайт

Ни для кого не секрет, что многие якобы независимые тесты и обзоры аудиоаппаратуры в специализированных журналах являются заказными. В этой задаче рассматривается подход к выставлению оценок, позволяющий журналистам создавать видимость объективного тестирования и при этом уравнивать суммарные оценки всех тестируемых устройств, тем самым не портя отношения с производителями.

Дана матрица $N \times M$ чисел: i-я строка матрицы представляет собой первоначальные оценки устройства под номером i по каждому из M критериев. За одну операцию журналист может отнять 1 от некоторой ячейки матрицы, прибавив при этом 1 к другой ячейке. Необходимо подсчитать минимальное число операций K, которые нужно произвести, чтобы сумма чисел во всех строках матрицы была одинаковой, и сумма чисел во всех столбцах матрицы тоже была одинаковой. Гарантируется, что для этого потребуется не более 10^5 операций. В процессе выполнения операций числа в матрице могут становиться отрицательными.

Input

В первой строке заданы размеры матрицы N и M ($1 \le N, M$; $N \times M \le 10^5$). Каждая из следующих N строк содержит по M чисел — i-ю строку матрицы A ($0 \le A_{i,j}$). Сумма всех чисел в матрице A не превосходит 10^9.

Output

В первой строке выведите искомое число операций K. В каждой из следующих K строк выведите по 4 числа — координаты клетки $(X_{i,1}; Y_{i,1})$, из которой нужно отнять 1, и координаты клетки $(X_{i,2}; Y_{i,2})$, в которую нужно добавить 1. Координаты должны удовлетворять условиям $1 \le X_{i,1..2} \le N$ и $1 \le Y_{i,1..2} \le M$.

Examples

stdin	stdout
2 3 1 6 5 7 2 3	0
4 4 1 1 2 2 1 0 1 1 0 0 1 1 1 1 1 2	3 1 3 2 1 1 4 3 2 4 4 3 2

Problem J. Педали эффектов

Input file:	stdin
Output file:	stdout
Time limit:	1 second
Memory limit:	256 megabytes

Некоторые гитаристы буквально помешаны на использовании различных педалей эффектов. Несколько педалей, подключенных последовательно, позволяют гитаристам добиваться уникального звучания.

Одному гитаристу, который провел целый день, экспериментируя с подключением педалей эффектов, наконец-то удалось добиться звучания своей мечты. Конечно, он тут же решил сообщить друзьям о своём достижении и пригласил их на прослушивание. Однако пока он разговаривал по телефону, младший брат решил поиграть на его гитаре...

Увидев несколько педалей, подключенных последовательно, часть из которых находилась в выключенном состоянии и пропускала через себя звук без каких-либо преобразований, младший брат в качестве эксперимента включил их все. Но стоило ему только дотронуться до струн, как в комнату прибежал разъяренный старший брат и обвинил его в том, что он испортил ему конфигурацию подключения педалей.

Младший брат, чтобы разрядить обстановку, сразу же заверил старшего в том, что он не менял порядок подключения педалей, а только включил их все. Кроме того, он запомнил, что до его вмешательства было включено некоторое ненулевое число педалей, и при этом не были включены никакие две педали подряд.

Чтобы убедить старшего брата в том, что восстановить исходную конфигурацию просто, младшему брату необходимо подсчитать, сколько существует различных конфигураций подключения педалей, удовлетворяющих описанным условиям.

Каждая конфигурация представляет собой некоторую подпоследовательность педалей, находящихся во включенном состоянии. Две конфигурации считаются одинаковыми, если педали в них совпадают в каждой из позиций. Педали, которые находятся в выключенном состоянии, не влияют на звук и не учитываются при сравнении конфигураций.

Input

В первой строке задается последовательность подключения педалей эффектов S, состоящая из строчных латинских букв ($1 \le |S| \le 10^5$). Педали эффектов, обозначаемые одной и той же буквой и находящиеся в разных позициях, являются идентичными.

Output

Искомое число различных конфигураций подключения педалей по модулю $10^9 + 7$.

Examples

stdin	stdout
abcc	5

Note

В первом тестовом примере подходят следующие конфигурации педалей: a, b, c, ac, bc.

Problem K. Случайный порядок

Input file:	stdin
Output file:	stdout
Time limit:	1 секунда
Memory limit:	256 мегабайт

Чаще всего аудиофилы предпочитают слушать музыкальные альбомы целиком, не меняя порядок следования треков. Считается, что любое нарушение задуманного исполнителем порядка мешает восприятию музыки.

Но сегодня аудиофила постигло жестокое разочарование — в книге, посвященной его любимой рок-группе, он прочитал, что порядок песен одного из альбомов при издании был изменен продюсером, причем без согласования с авторами.

Эта история так расстроила аудиофила, что отныне он решил слушать треки этого альбома только в случайном порядке. Для этого он будет использовать операцию перемешивания на своем плеере.

Операция перемешивания в плеере работает по следующим правилам:

1) Позиция каждого трека после перемешивания должна отличаться от его позиции до перемешивания.

2) Если до перемешивания два трека находились на соседних позициях, после перемешивания они не должны располагаться на соседних позициях.

Ваша задача — по заданному числу композиций N вывести любой порядок следования треков альбома, удовлетворяющий правилам перемешивания, при том, что изначально все треки пронумерованы числами от 1 до N.

Input

Число треков на альбоме N ($1 \le N \le 10^5$).

Output

В N строках выведите номера треков после перемешивания, по одному в каждой строке, согласно условию. Если существует несколько решений — выведите любое. В случае, если не существует порядка треков, который бы удовлетворял правилам перемешивания, в единственной строке выведите «No solution» (без кавычек).

Examples

stdin	stdout
2	No solution
4	2 4 1 3