

Problem A. Интернет-зависимость

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

Одной из самых распространенных форм интернет-зависимости считается бесконечный серфинг по сети без какой-либо осознанной причины. В принципе, в этом нет ничего нового для человека — такая привычка мало чем отличается, например, от листания каналов телевизора. Однако в случае обрыва ТВ-кабеля мало кому приходит в голову считать пылинки на экране или вслушиваться в звук помех. Некоторые интернет-пользователи, напротив, после обрыва соединения проводят длительное время за изучением открытой веб-страницы, даже если она не несет никакой новой информации.

В браузере отображена страница, содержащая некоторую строку S , состоящую из строчных латинских букв. Для поиска текста на странице используется специальное окно, в процессе ввода строки в которое подсвечиваются все её вхождения на текущей странице. Ваша задача — подсчитать минимальное число действий, которое необходимо осуществить, чтобы каждый символ строки S был подсвечен хотя бы один раз.

Разрешенные действия:

- 1) набор одной строчной латинской буквы;
- 2) стирание последней набранной буквы.

Input

В первой строке записано единственное число N ($1 \leq N \leq 10^5$).

Во второй строке задается строка S длины N , состоящая из строчных букв латинского алфавита.

Output

Искомое число действий.

Examples

| stdin | stdout |
|-------------|--------|
| 6 ababab | 2 |
| 4 abac | 4 |

Note

В первом тестовом примере после ввода строки «a» подсвечиваются все символы «a». После ввода «b» (без стирания «a») подсвечиваются все строки «ab». Таким образом, за два действия подсвеченными хотя бы один раз оказываются все символы строки.

Во втором тестовом примере предлагается следующая последовательность действий:

1. Ввод символа «a» — подсвечиваются все вхождения «a».
2. Ввод символа «b» — подсвечиваются все вхождения «ab».
3. Стирание символа «b» — подсвечиваются все вхождения «a».
4. Ввод символа «c» — подсвечиваются все вхождения «ac».

Все символы строки оказываются подсвеченными хотя бы один раз за 4 действия.

Problem B. 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 ($1 \leq i \leq R - L + 1$) 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 ($1 \leq K \leq R - L + 1$), 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 ($0 \leq L \leq R \leq 10^{18}$, $0 \leq X \leq 10^{18}$, $1 \leq K \leq R - L + 1$).

Output

Исходный номер выигрышного билета.

Examples

| <code>stdin</code> | <code>stdout</code> |
|--------------------|---------------------|
| 6 15 0 7 | 12 |
| 1 20 10 5 | 14 |

Problem C. 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 ($1 \leq A, B \leq 10^9$) — квалификация первого и второго игрока, соответственно.

Output

Общее число подтягиваний, совершаемых за время игры обоими игроками.

Examples

| <code>stdin</code> | <code>stdout</code> |
|--------------------|---------------------|
| 3 5 | 18 |
| 1 2 | 8 |

Problem D. 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, \dots, 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 $1 \leq i \leq k-1$), 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 ($1 \leq i \leq k$), что $X_i < Y_i$, и для всех j ($1 \leq j < i$) 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, \dots, 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 ($2 \leq N \leq 10^3$) и M ($1 \leq M \leq 10^4$).

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

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

Далее в R строках задается описание маршрутов, по одному на каждую строку. Описание i -го маршрута содержит число k_i ($2 \leq k_i \leq N$) — число опорных перекрестков i -го маршрута и k_i чисел $A_{i,j}$ ($1 \leq A_{i,j} \leq N$) — 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 | 644 |
| 1 2 | 644#2 |
| 2 3 | 211 |
| 3 4 | 40 |
| 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 | |

Problem E. Шахматы вслепую

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

Чтобы научиться играть в шахматы вслепую, нужно не только постоянно практиковаться, но и выполнять специальные упражнения.

Дана шахматная доска размером $N \times M$ клеток. Необходимо подсчитать, сколько клеток этой доски достижимы из клетки $(X; Y)$ хотя бы одной из шахматных фигур так, чтобы число ходов для выбранной фигуры было минимально возможным и при этом нечетным.

Input

В первой строке записаны 4 целых числа N, M, X, Y — размер шахматного поля и координаты начальной клетки ($1 \leq N, M \leq 10^9, 1 \leq X \leq N, 1 \leq Y \leq M$).

Output

Искомое число клеток.

Examples

| stdin | stdout |
|---------|--------|
| 5 5 3 3 | 24 |
| 8 8 2 3 | 57 |

Problem F. Генеральские часы

Input file: `stdin`
Output file: `stdout`
Time limit: 4 seconds
Memory limit: 256 megabytes

Генерал, приехавший в одну из военных частей с ревизией, неспешно прогуливался по взлетно-посадочной полосе с командиром части, обсуждая боеспособность армии и политическую обстановку в мире. Вернувшись с прогулки, генерал обнаружил пропажу карманных наградных часов, подаренных ему за особые заслуги еще в молодости. Итак, над командиром части нависла угроза досрочного выхода на пенсию, если его солдаты не найдут генеральские часы в самое ближайшее время.

Взлетная полоса имеет протяженность K метров. Всего в части служит N солдат, и каждый из них регулярно участвует в прополке взлетно-посадочной полосы, отвечая за некоторый участок, начинающийся в A_i и заканчивающийся в B_i метрах от начала полосы.

Командиру части некогда подбирать оптимальный состав поисковой группы, а отправлять всю часть на поиски может быть слишком рискованным мероприятием. Поэтому он построил всех солдат в шеренгу по одному человеку и решил отправить на поиски часов только группу из нескольких стоящих подряд солдат. При этом каждый солдат поисковой группы будет обследовать только тот участок полосы, за который он отвечает при прополке.

По окончании поисков солдаты должны установить специальные флажки в начале и конце каждого непрерывного участка взлетно-посадочной полосы, в котором работал хотя бы один солдат. Таким образом, если участки нескольких солдат группы поиска пересекаются или касаются друг друга, образуя один связный участок, устанавливаются только два флажка на его границах. Если участок некоторого солдата не пересекается и не касается ни с одним из участков других солдат из группы поиска, то на его границах также устанавливаются два флажка.

Со снабжением флажками в части дела обстоят не лучшим образом, и поэтому, прежде чем отправить некоторую группу солдат на поиски, командир части хочет знать, сколько ей потребуется флажков для разметки взлетно-посадочной полосы.

Input

В первой строке заданы два числа N ($1 \leq N \leq 10^4$) и K ($1 \leq K \leq 10^3$) — количество солдат и длина взлетно-посадочной полосы. Следующие N строк содержат описания участков взлетно-посадочной полосы, за прополку которых отвечают солдаты. В $(i + 1)$ -й строке задаётся описание участка прополки, закрепленного за i -м солдатом, в виде двух целых чисел A_i и B_i ($0 \leq A_i < B_i \leq K$).

В следующей строке задано число M ($1 \leq M \leq 10^5$) — количество запросов. В каждой из следующих M строк содержится описание запроса в виде двух чисел L_i и R_i ($1 \leq L_i \leq R_i \leq N$) — номера первого и последнего солдата поисковой группы.

Output

Выведите M строк с ответами на запросы: i -я строка должна содержать единственное число — количество флажков, которые потребуются для разметки взлетно-посадочной полосы поисковой группе, описанной в i -м запросе.

Examples

| stdin | stdout |
|-------|--------|
| 8 10 | 2 |
| 2 7 | 4 |
| 0 3 | 2 |
| 4 8 | 6 |
| 3 4 | 2 |
| 5 6 | |
| 7 8 | |
| 4 5 | |
| 6 7 | |
| 5 | |
| 1 2 | |
| 2 3 | |
| 1 3 | |
| 4 6 | |
| 4 8 | |

Problem G. Вахтер

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

Работать вахтером в студенческом общежитии — это настоящее 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 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 ($1 \leq N \leq 1000$) и L ($1 \leq L \leq 1000$) — количество дверных хлопков и предел терпения вахтера. Во второй строке задано N чисел A_i ($1 \leq A_i \leq 1000$) — громкости дверных хлопков.

Output

Число прочитанных за день нотаций.

Examples

| <code>stdin</code> | <code>stdout</code> |
|--------------------|---------------------|
| 6 3 1 3 2 2 3 2 | 3 |

Problem H. 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 ($1 \leq N \leq 10^9$).

Output

Искомое число операций.

Examples

| <code>stdin</code> | <code>stdout</code> |
|--------------------|---------------------|
| 3 | 3 |

Problem I. 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 ($1 \leq N \leq 10^5$). В следующих $N - 1$ строках задаются пары чисел U_i и V_i ($1 \leq U_i, V_i \leq N$) — номера остановок, которые соединяет i -й участок пути.

В следующей строке задано число запросов M ($1 \leq M \leq 10^5$). Следующие M строк содержат описания запросов в виде троек чисел A_i , B_i и C_i ($1 \leq A_i, B_i, C_i \leq N$), 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 |
| 1 2 | 1 |
| 6 2 | 2 |
| 3 1 | |
| 3 4 | |
| 3 5 | |
| 3 | |
| 1 2 3 | |
| 6 4 5 | |
| 4 3 2 | |

Problem J. 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 K. Чужая диссертация

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

Некоторые аспиранты долго не могут определиться, чему будет посвящено их диссертационное исследование. Когда терпение научного руководителя подходит к концу, им ничего не остается, как отправиться в университетскую библиотеку, взять первую попавшуюся работу по своему направлению и, внося минимальные изменения, попытаться выдать большую часть текста за собственные наработки. Вариант беспроигрышный — если даже основная часть будет забракована руководителем за отсутствие научной новизны, то хотя бы введение и обзор предметной области не придется делать с нуля.

Одному из таких аспирантов попала в руки чья-то диссертация, посвященная исследованию сигма-функции Вейерштрасса.

Коэффициенты сигма-функции Вейерштрасса заданы соотношением:

$A_{i,j} = 0$ если $i < 0$ или $j < 0$;

$A_{0,0} = 1$;

$A_{i,j} = 3(i+1) * A_{i+1,j-1} + \frac{16(j+1)}{3} A_{i-2,j+1} - \frac{(3j+2i-1)*(6j+4i-1)}{3} A_{i-1,j}$ в остальных случаях.

Диссертация была написана ещё в 80-е годы, когда о современной производительности вычислительных машин можно было только мечтать. Тем не менее её автору удалось получить несколько таблиц.

В первой таблице представлены значения $A_{i,j}$ для $0 \leq i, j$ и $i + j \leq 5$.

$$\left(\begin{array}{cccccc} 1 & -1 & -9 & 69 & 321 & 160839 \\ -3 & -18 & 513 & 33588 & 2808945 & \\ -54 & 4968 & 257580 & 20019960 & & \\ 14904 & 502200 & 162100440 & & & \\ 1506600 & 796330440 & & & & \\ 2388991320 & & & & & \end{array} \right)$$

Во второй таблице представлены показатели степени 2-х в разложении $A_{i,j}$ на простые множители для $0 \leq i, j$ и $i + j \leq 20$.

| | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 2 | 0 | 1 | 0 | 3 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 4 | 0 | 1 | 0 | 2 | 0 |
| 1 | 3 | 2 | 3 | 1 | 4 | 3 | 4 | 1 | 3 | 2 | 3 | 1 | 5 | 4 | 5 | 1 | 3 | 2 | 0 | 0 |
| 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 3 | 3 | 0 | 0 | 0 |
| 3 | 3 | 5 | 5 | 4 | 4 | 5 | 5 | 3 | 3 | 6 | 6 | 5 | 5 | 6 | 6 | 3 | 0 | 0 | 0 | 0 |
| 3 | 6 | 5 | 6 | 4 | 6 | 5 | 6 | 3 | 7 | 6 | 7 | 5 | 7 | 6 | 7 | 0 | 0 | 0 | 0 | 0 |
| 6 | 7 | 6 | 8 | 6 | 7 | 6 | 10 | 7 | 8 | 7 | 9 | 7 | 8 | 7 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 7 | 8 | 8 | 7 | 7 | 10 | 10 | 8 | 8 | 9 | 9 | 8 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 7 | 7 | 7 | 9 | 9 | 9 | 9 | 8 | 8 | 8 | 8 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 8 | 7 | 11 | 9 | 10 | 9 | 11 | 8 | 9 | 8 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 12 | 11 | 12 | 10 | 12 | 11 | 12 | 9 | 12 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 12 | 13 | 13 | 12 | 12 | 14 | 14 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 14 | 13 | 14 | 12 | 15 | 14 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 15 | 14 | 17 | 15 | 16 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 15 | 17 | 17 | 16 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 15 | 15 | 15 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 16 | 15 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 18 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

В третьей таблице представлены показатели степени 3-х в разложении $A_{i,j}$ на простые множители для $0 \leq i, j$ и $i + j \leq 20$.

| | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 0 | 2 | 1 | 1 | 3 | 2 | 4 | 5 | 4 | 4 | 6 | 5 | 5 | 8 | 7 | 8 | 9 | 8 | 8 | 12 |
| 1 | 2 | 3 | 3 | 3 | 6 | 5 | 5 | 7 | 5 | 6 | 7 | 8 | 8 | 10 | 9 | 9 | 11 | 9 | 12 | 0 |
| 3 | 3 | 5 | 4 | 6 | 7 | 7 | 7 | 9 | 7 | 7 | 10 | 9 | 10 | 11 | 11 | 11 | 15 | 13 | 0 | 0 |
| 4 | 4 | 7 | 6 | 6 | 8 | 7 | 8 | 9 | 9 | 9 | 11 | 10 | 10 | 12 | 11 | 14 | 15 | 0 | 0 | 0 |
| 5 | 7 | 8 | 8 | 8 | 10 | 9 | 9 | 12 | 10 | 11 | 12 | 12 | 12 | 16 | 15 | 15 | 0 | 0 | 0 | 0 |
| 8 | 8 | 10 | 9 | 10 | 11 | 12 | 12 | 14 | 12 | 12 | 14 | 13 | 16 | 17 | 17 | 0 | 0 | 0 | 0 | 0 |
| 9 | 9 | 11 | 10 | 10 | 13 | 12 | 13 | 14 | 13 | 13 | 17 | 16 | 16 | 18 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 11 | 12 | 13 | 13 | 15 | 14 | 14 | 16 | 14 | 17 | 18 | 18 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 12 | 15 | 14 | 15 | 16 | 16 | 16 | 20 | 18 | 18 | 20 | 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 13 | 15 | 14 | 14 | 16 | 15 | 18 | 19 | 18 | 18 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 15 | 16 | 16 | 16 | 20 | 19 | 19 | 21 | 19 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 16 | 18 | | | | | | | | | | | | | | | | | | |

Output

Для каждого теста из входных данных в отдельной строке выведите два числа: искомые показатели степени 2-х и 3-х. При этом если $A_{n,m}$ не является целым числом, необходимо вывести строку «-1 -1» (без кавычек).

Examples

| stdin | stdout |
|-------|--------|
| 3 | 5 10 |
| 2 5 | 1 2 |
| 1 1 | 10 19 |
| 15 7 | |