

Problem A. Три маяка

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

*Если есть на свете что-либо
надежное, так это свет маяка.
(Тове Янссон)
Глушь такая, что хочется
определить широту и долготу.
(Из письма О.Э.Мандельштама)*

Шхуна «Победа» и её легендарный капитан Христофор Бонифатьевич Врунгель отправились на международную парусную регату. Не имея специальных инструментов для определения координат, шхуна сбилась с курса и заблудилась. К счастью, у капитана Врунгеля есть бинокль, и в бинокль хорошо видны 3 маяка, координаты которых — опять же, к счастью для капитана — хорошо известны. Как известно, с помощью бинокля можно не только наблюдать объекты, но также измерять их угловые размеры, то есть углы между прямыми линиями, соединяющими крайние точки рассматриваемых объектов и глаз наблюдателя. Например, если два маяка расположены в точках M_1 и M_2 , а шхуна — в точке K , то угловой размер отрезка M_1M_2 , соединяющего эти маяки, — это угол M_1KM_2 .

Вам необходимо определить координаты шхуны по известным значениям угловых размеров отрезков M_1M_2 и M_2M_3 , соединяющих маяки M_1 и M_2 , M_2 и M_3 .

Input

В первой строке записано одно целое число n — количество возможных положений маяков ($1 \leq n \leq 50\,000$). В каждой из n следующих строк записаны восемь целых чисел, из них первые шесть чисел — декартовы координаты различных точек M_1 , M_2 , M_3 , не превосходящие по модулю 10 000; следующие два целых числа из промежутка $[1; 179]$ — угловые размеры (в градусах) отрезков M_1M_2 и M_2M_3 . Для каждого расположения маяков гарантируется, что существует хотя бы одно решение.

Output

Для каждого расположения маяков выведите через пробел координаты любого из возможных положений шхуны. Ответ считается верным, если абсолютная или относительная погрешность не превосходит 10^{-6} .

Examples

| стандартный ввод | стандартный вывод |
|------------------------|---------------------------|
| 1 0 0 2 0 2 2 90 90 | 1.0000000000 1.0000000000 |

Problem B. ВСПД 1.0

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

Министерство инноваций утвердило приказ о построении ВСПД, объединяющей две столицы. Все, конечно, поняли, что речь о высокоскоростной пешеходной дороге, но никто не знал, как её строить. К счастью, в городе нашелся единственный в мире завод по производству комплектующих для СПД.

Завод производит m типов плиток в форме прямоугольников размеров $1 \times k$ см, доступных в c_k различных цветах, $1 \leq k \leq m$.

Дорога состоит из прямоугольных сегментов размера $2 \times n$ см. Каждый сегмент при строительстве собирается из плиток указанного вида. Поскольку готовую дорогу будет принимать высокое начальство, решено её строить из уникальных сегментов, то есть все сегменты должны отличаться способом покрытия плиткой.

Теперь важно выяснить, каково максимальное возможное количество уникальных сегментов.

Input

В первой строке записаны целые числа n и m — длина сегмента и количество типов плиток ($1 \leq n \leq 10^{18}$, $1 \leq m \leq 9$). Вторая строка содержит записанные через пробел m целых неотрицательных чисел c_1, c_2, \dots, c_m , среди которых хотя бы одно отлично от нуля. Все числа не превышают 10^9 .

Output

Выведите искомое количество по модулю $(10^9 + 9)$.

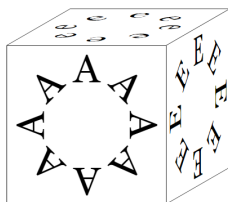
Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 3 0 1 2 | 7 |
| 2 4 0 0 1 0 | 0 |

Problem C. Кубики

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

Геннадий, всю жизнь проработавший на казанском заводе игровых автоматов, поехал в Лас-Вегас и в первом же казино обнаружил автомат своего производства. Этот автомат содержит N кубиков, на каждой грани которых особым образом написаны буквы латинского алфавита.



В начале игры автомат выставляет кубики друг на друга так, что игроку видны только буквы на одной из сторон башни. Далее на экране отображается слово из N букв, которое игрок должен составить на видимой стороне башни, вращая кубики вокруг вертикальной оси. Слово читается сверху вниз.

Гена имеет огромное преимущество — он помнит все кубики, произведенные на его заводе, и может отличить их друг от друга в автомате. Однако, он не может знать точную ориентацию кубика в башне, так как видит только одну грань.

Помогите Геннадию вычислить вероятность того, что поставленная автоматом задача выполнима.

Input

В первых двух строках записаны слова, длиной N символов — начальное и целевое слова на видимой стороне башни кубиков ($1 \leq N \leq 10$). Следующие N строк содержат по 6 символов описывающих буквы на гранях кубиков в следующем порядке: лицевая, верхняя, правая, нижняя, левая, задняя. Кубики перечислены по порядку от верхнего к нижнему. Заглавные и строчные буквы считаются различными.

Output

Вывести искомую вероятность. Ответ считается верным, если относительная или абсолютная погрешность не превышает 10^{-6} .

Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| HALLW | 0.25 |
| HELLO | |
| XABCDH | |
| XAECDe | |
| AbcdeL | |
| AbcdeL | |
| ABOWCD | |

Note

На втором кубике мы видим грань с буквой «А», и нам требуется повернуть кубик так, чтобы была видна грань с буквой «Е». В зависимости от поворота грани с буквой «А», грань с буквой «Е» может находиться как «сбоку», так или «сверху» или «снизу» по отношению к Геннадию. В первом случае кубик можно повернуть требуемым образом, а во втором и третьем — нет. Вероятность нахождения «Е» «сбоку» равна вероятности нахождения «сверху» или «снизу» и равна $1/2$. Поэтому вероятность повернуть этот кубик правильным образом равна $1/2$. Аналогично и с буквами «W» и «O», поэтому общая вероятность равна $1/4$.

Problem D. Верблогистика

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

Логистика — сложнейшая наука. Логистика в пустыне — невероятное искусство.

В бескрайней пустыне расположен склад, в котором хранятся N яблок. В L километрах от яблочного склада находится завод, производящий яблочный сок. Единственным средством транспортировки яблок является грузовой верблюд, припаркованный около склада. Верблюд обладает феноменальной грузоподъемностью, но не может нести более K яблок. Кроме того, верблюд должен съесть по одному яблоку пройдя каждый километр, иначе следующую партию везти будет уже некому. Зато на пути от склада к заводу через каждый километр установлены навесы, где можно оставлять любое количество яблок.

Выясните, какое максимальное количество яблок возможно доставить на завод.

Input

В единственной строке через пробел записаны целые числа L, N, K ($1 \leq L, N, K \leq 10^9$).

Output

Вывести единственное число — искомое количество яблок.

Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 30 10 | 11 |
| 1000 2999 1001 | 533 |
| 5 4 10 | 0 |

Problem E. Ковер

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

На отдаленной планете обнаружен кратер идеально круглой формы. Для изучения кратера на поверхность спущены две стационарные станции. Для организации научного процесса необходимо проложить по поверхности особую ковровую дорожку, по которой будут перемещаться планетоходы. Ковровую дорожку можно прокладывать по всей поверхности, кроме территории, занятой кратером. Необходимо, чтобы по дорожке можно было добраться от одной станции до другой и до границы кратера.

Ковры в условиях космоса — невероятно дорогой ресурс, поэтому необходимо минимизировать длину ковровой дорожки.

Input

В первой строке записаны две пары чисел, разделенных пробелами — декартовы координаты станций. Во второй строке — координаты центра и радиус кратера. Все числа целые, по модулю не превышающие 40000. Гарантируется, что станции не находятся внутри кратера.

Output

Вывести общую длину необходимой ковровой дорожки с абсолютной или относительной погрешностью, не превышающей 10^{-6} .

Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 4 4 4 0 0 5 | 1.000000 |

Problem F. НОД и НОК

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

*Вещи бывают великими и малыми не
только по воле судьбы и
обстоятельств, но также по
понятиям каждого.*
(Козьма Прутков)

Вам приходилось когда-нибудь вычислять наибольший общий делитель нескольких чисел? А наименьшее общее кратное? Ну, конечно, да... А приходилось ли вам находить сами числа по известным значениям их наибольшего общего делителя (НОД) и наименьшего общего кратного (НОК)? Ну или хотя бы определять, сколько таких наборов имеют заданные НОД и НОК? Наверное, нет...

Итак, вам необходимо подсчитать количество упорядоченных наборов из k целых положительных чисел, у которых наибольший общий делитель и наименьшее общее кратное равны d и m соответственно. Например, при $k = 2$, $d = 2$, $m = 12$ таких упорядоченных наборов четыре: $(2, 12)$, $(12, 2)$, $(4, 6)$ и $(6, 4)$.

Input

В первой строке записано одно целое число k — количество чисел в наборах ($2 \leq k \leq 10^{18}$). Во второй строке записаны целые числа d и m — наибольший общий делитель и наименьшее общее кратное чисел набора ($1 \leq d \leq m \leq 10^9$).

Output

Выведите количество упорядоченных наборов по модулю $(10^9 + 9)$.

Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 2 12 | 4 |
| 3 4 5 | 0 |

Problem G. Кастрюли

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

В магазине кастрюль резко упали продажи. Маркетологи магазина провели исследования и поняли, что причина в сковородках. Люди перестали покупать кастрюли, так как сковородки и дешевле, и компактнее при хранении. Совет директоров магазина принял решение расширить ассортимент и начать продажи сковородок. Первая партия уже заказана.

Отделу складской логистики поручено подготовить место под новый товар. Сейчас на складе размещены N кастрюль, для каждой из которых известен диаметр D_i . Существует единственный способ экономии пространства — можно в любую кастрюлю вложить ровно одну кастрюлю меньшего диаметра, в которую могут быть вложены другие.

Помогите логисту найти минимальное количество кастрюль на складе, в которые можно будет вложить все остальные.

Input

В первой строке записано единственное число N ($1 \leq N \leq 1000$). Во второй строке — N целых чисел D_i , разделенных пробелом ($1 \leq D_i \leq 10\,000$).

Output

Выведите искомое число.

Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 7 5 2 5 2 | 2 |

Problem H. Мессенджер

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

Новые технологии появляются постоянно. Несколько лет назад был запущен знаменитый сервис обмена сообщениями под названием «LCI». Всемирную известность мессенджер приобрел благодаря технической ошибке — все узлы связи искажали сообщения. Каждому узлу связи был назначен числовой ключ X_i , и, транспортируя сообщение, узел инвертировал его префикс длиной X_i символов и суффикс длиной $L - X_i$ символов, где L — длина сообщения. Текст становился неузнаваемым, однако пользователям это нравилось.

Недавно вышла версия 16.0, в которой разработчики наконец-то исправили досадную ошибку. Сообщения стали доходить без искажений, но оказалось, что таким образом «LCI» стал таким же скучным, как сотни других мессенджеров. Пользователи потеряли всякий интерес к системе и начали удалять учетные записи.

Разработчики мессенджера пытались вернуть пользователям любимившуюся странность, но выяснилось, что откатить узлы связи на старую прошивку нельзя. Теперь требуется создать эмулятор сети «LCI», который сможет по отправленному сообщению и ключам пройденных узлов вычислить конечное сообщение.

Input

В первой строке записано отправленное сообщение — строка длиной L символов ($1 \leq L \leq 200\,000$), состоящая из строчных букв латинского алфавита, цифр, пробелов и знаков препинания. Во второй строке единственное число N ($1 \leq N \leq 200\,000$) — количество узлов связи, пройденных сообщением. Следующие N строк содержат ключи узлов связи X_i ($0 \leq X_i \leq L$) в порядке их прохождения.

Output

Единственная строка должна содержать конечное сообщение.

Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| abcde | baedc |
| 5 | |
| 2 | |
| 3 | |
| 5 | |
| 3 | |
| 1 | |

Problem I. Манхэттенский проект

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 4 секунды
Memory limit: 256 мегабайт

В семимерном пространстве существует шестимерное учреждение, где пятимерные сотрудники ведут четырехмерную базу данных. В базе данных записана информация о точках, заданных четырьмя координатами. В учреждение приходят три типа запросов:

1. Добавить точку с указанными координатами.
2. Удалить точку с указанными координатами.
3. Предоставить расстояние от заданной точки до наиболее удаленной от неё.

При этом в данном пространстве расстояние между точками (x_1, x_2, x_3, x_4) и (y_1, y_2, y_3, y_4) вычисляется по формуле $|x_1 - y_1| + |x_2 - y_2| + |x_3 - y_3| + |x_4 - y_4|$.

В учреждении грядет сокращение штата, и назрела необходимость автоматизировать описанную работу.

Input

В первой строке целое число N ($1 \leq N \leq 10^5$) — количество запросов. В следующих N строках по пять целых чисел: t, x_1, x_2, x_3, x_4 — тип запроса и четыре координаты точки соответственно ($1 \leq t \leq 3, -10^8 \leq x_1, x_2, x_3, x_4 \leq 10^8$).

Гарантируется, что на момент запроса типа 1 (добавление) точки (x_1, x_2, x_3, x_4) не существует, на момент запроса типа 2 (удаление) точка (x_1, x_2, x_3, x_4) существует, на момент запроса типа 3 (запрос расстояния) хотя бы одна точка существует. До первого запроса в базе нет ни одной точки.

Output

Для каждого запроса типа 3 вывести в отдельной строке результат запроса.

Examples

| стандартный ввод | стандартный вывод |
|--|-------------------|
| 5 1 0 0 0 0 1 -10 2 6 -9 3 -8 0 9 -5 2 0 0 0 0 3 -8 0 9 -5 | 22 11 |
| 9 1 2 0 0 0 1 4 3 0 0 1 1 5 0 0 3 2 3 0 0 3 -1 2 0 0 2 4 3 0 0 3 -1 2 0 0 2 2 0 0 0 3 1 5 0 0 | 3 6 5 0 |

Problem J. Жижя

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 2 секунды
Memory limit: 512 мегабайт

Шел 2316 год. Человечество, вдохновленное успешной колонизацией XMars'a, приступило к колонизации соседней планеты — XVenus. По традиции сначала к планете отправили автоматическую станцию. Её цель — подготовить площадки и их окрестности для размещения жилых модулей. Для этого со станции, вращающейся на орбите планеты, вниз опускается небольшой корабль с запасом жижжи особого состава и выливает её на поверхность планеты. Места, обработанные данным составом, становятся пригодными для жизни.

Логически процесс организован следующим образом. Поверхность планеты разделена на множество секторов, пронумерованных от нуля. Программа станции описывает n этапов. Этапы бывают двух типов: либо обработка поверхности жижжей, либо очистка поверхности от жижжи. Если на i -ом этапе происходит обработка поверхности, то корабль берет с собой p_i тонн жижжи. Одной тонны жижжи хватает на обработку одного сектора. Корабль сканирует поверхность, находит первый необработанный сектор с номером не менее x_i и, начиная с этого сектора, обрабатывает секторы поочередно в порядке возрастания номера. Если состав закончится или на пути встретится уже обработанный сектор, то корабль прекращает работу и поднимается на станцию, возвращая запас жижжи в общее хранилище. Далее корабль снова заправляется и вылетает для выполнения следующего этапа.

Очистка поверхности от жижжи придумана программистами как «костыль» для исправления ошибок обработки планеты. Если на i -ом этапе осуществляется очистка от жижжи, то корабль спускается на сектор с номером x_i и, если он залит жижжей, то очищает его, восстанавливая первоначальное состояние.

Аналитикам команды колонизаторов необходимо для каждого этапа первого типа вычислить номера секторов, которые будут обработаны.

Input

В первой строке записано целое положительное число n , не превосходящее 10^5 . В последующих n строках описаны этапы. Этап обработки жижжей описывается двумя целыми положительными числами x_i и p_i , записанными через пробел ($0 < x_i \leq 5 \cdot 10^8$, $0 < p_i \leq 1000$). Этап очистки от жижжи описывается единственным целым отрицательным числом y_i ($-5 \cdot 10^8 \leq y_i < 0$). Это означает, что нужно очистить сектор с номером $x_i = -y_i$.

Output

Для каждого этапа обработки жижжей в отдельной строке вывести пару чисел, разделенных пробелом: минимальный и максимальный номера секторов, обработанных в рамках этого этапа. Результаты этапов выводить в порядке следования во входном файле.

Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6 | 5 6 |
| 5 2 | 7 9 |
| 6 3 | 4 4 |
| 4 7 | 1 1 |
| 1 1 | 2 4 |
| -4 | |
| 2 4 | |

Problem K. Синонимичная система счисления слов

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 0.5 секунды
Memory limit: 256 мегабайт

Синонимичная система счисления слов (СССС) оперирует строками из N символов «0» и «1» ($2 \leq N \leq 10000$). В рамках данной системы счисления существует единственная операция над словами — синонимизация. Синонимизацией называется замена порядка символов в любой подстроке слова, количество единиц в которой четно, на обратный.

Если из одного слова можно получить другое, совершив не более $16 \cdot N$ синонимизаций, то такие слова называются синонимами.

Необходимо выяснить, являются ли заданные слова синонимами, и, если являются, привести подходящую последовательность синонимизаций.

Input

В двух строках записаны разные слова длиной N символов, по одному в каждой строке.

Output

Если заданные слова не являются синонимами, то в единственной строке вывести «NO».

В противном случае, в первой строке вывести «YES». Во второй строке число K ($K \leq 16 \cdot N$) — количество синонимизаций, необходимых для преобразования. В следующих K строках — пары чисел, описывающих преобразование первого слова во второе. Первое число задает номер первого символа, участвующего в синонимизации, второе число (всегда больше первого) — номер последнего символа.

Examples

| стандартный ввод | стандартный вывод |
|------------------------|-------------------|
| 110001111 111100011 | YES 1 3 7 |
| 110 010 | NO |

Problem L. Три автомата

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

Три автомата считывают числа на вставленных карточках и печатают на новых карточках пары натуральных чисел.

Первый автомат, прочитав карточку $(a; b)$, выдаёт новую карточку $(a + 1; b + 1)$. Второй автомат, прочитав карточку $(a; b)$, в случае если оба числа четные, выдаёт карточку $(a/2; b/2)$. В третий автомат нужно вставлять пару карточек $(a; b)$ и $(b; c)$, в ответ он выдаёт карточку $(a; c)$. Автоматы возвращают все прочитанные карточки.

Имеется единственная карточка $(a; b)$. Необходимо получить карточку $(c; d)$ не более чем за 15000 операций с автоматами, либо указать, что это невозможно.

Input

В первой строке записаны целые числа a и b . Во второй - c и d ($1 \leq a, b, c, d \leq 2000$). Начальная и конечная карточка различны.

Output

Если получить требуемую карточку невозможно, вывести 0. Иначе в первой строке вывести число k — количество использований автоматов. В дальнейших k строках вывести описание очередного использования автомата в формате « <номер автомата> <номера на первой карточке> [<номера на второй карточке>] ». В случае использования третьего автомата второй номер на первой карточке и первый номер на второй карточке должны быть одинаковыми.

Examples

| стандартный ввод | стандартный вывод |
|------------------|----------------------------------|
| 2 4 1 3 | 3 2 2 4 1 1 2 3 1 2 2 3 |
| 4 2 1 1 | 0 |

Problem M. Наименьшая дробь

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

*Три с половиною овечки
и восемь сотых пастуха
однажды встретили у речки
четыре пятых петуха.*
М. Вейцман «Действия с дробями».

В настоящее время миру известно огромное количество сенсационных археологических находок. Во время последних раскопок были обнаружены носители информации, относящиеся к эпохе программирования XX века. Расшифровка файлов позволила ученым доказать гипотезу о том, что древние программисты обладали искусством производить простейшие арифметические операции с дробями. Многие тексты были расшифрованы, многие загадки были разгаданы. Однако среди нерешенных оказалась задача о вычислении *наименьшей* положительной дроби, при делении которой на каждую из n заданных дробей получаются целые числа.

Возможно, это удастся сделать вам...

Input

В первой строке записано одно целое число n — количество заданных дробей ($1 \leq n \leq 6$). В каждой из следующих n строк записано два целых числа a_i, b_i — числитель и знаменатель несократимой дроби ($1 \leq a_i \leq 10^3, 1 \leq b_i \leq 10^9$).

Output

В единственной строке запишите через пробел два положительных целых числа — числитель и знаменатель наименьшей несократимой дроби, удовлетворяющей условию задачи.

Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 1 2 3 4 | 3 2 |
| 2 2 3 4 5 | 4 1 |