Problem A. Самая красивая

Input file:	`beautiful.in`
Output file:	`beautiful.out`
Time limit:	2 секунды
Memory limit:	256 мебибайт

«Ты прекрасна, спору нет;
Но царевна всех милее,
Всех румяней и белее»

А. С. Пушкин, «Сказка о мёртвой царевне и семи богатырях»

Пришло время ежегодного конкурса красоты! Только представьте шеренги красавиц, стремящихся показать себя с лучшей стороны... К сожалению, поля страницы слишком малы, чтобы вместить их фотографии.

Судьям конкурса предстоит адски тяжёлая работа. Споры отнимут несколько дней и не одну оторванную пуговицу. Чтобы упростить процедуру, глава судейского комитета предложил следующую процедуру.

Вначале каждый судья выписывает участниц в порядке его предпочтений. Первой в списке стоит девушка, которая меньше всего нравится судье, второй — наименее нравящаяся ему из оставшихся, и так далее. После этого судьи последовательно распределяют участниц по местам: первый судья выбирает девушку, которая займёт последнее место, второй выбирает вторую с конца и так далее. После того, как n-й судья выберет очередную из них, вновь приходит очередь первого. Процесс продолжается, пока все девушки не будут распределены по местам. Каждый судья выбирает первую (то есть наименее нравящуюся ему) девушку в своём списке, которая ещё не выбрана.

Вася — судья номер k. Он уже знает, в каком порядке каждый из судей расположил участниц (телепатия, знаете ли). И на самом деле ему важна только одна участница: Катя. Вася хочет расположить девушек в своём списке так, чтобы место, которое займёт Катя, было как можно выше. Помогите ему в этом сложном деле.

Input

Входные данные состоят из одного или более тестов.

Каждый тест начинается строкой, в которой записаны целые числа n ($1 \le n \le 1\,000$), определяющее число судей и k ($1 \le k \le 1\,000$), определяющее число участниц. Во второй строке записаны два числа: номер Васи среди судей v ($1 \le v \le n$) и номер Кати среди участниц x ($1 \le x \le k$). Следующие n строк содержат списки, которые составили судьи, по k чисел в каждом. Каждый список начинается с девушки, которую соответствующий судья поставил бы на последнее место. Вместо васиного списка в соответствующей строке записаны нули.

Входные данные завершаются тестом, где $n = k = 0$, который не нужно обрабатывать.

Суммы n и k по всем тестам во входных данных не превысят $1\,000$ каждая.

Output

Для каждого теста выведите самое высокое место, которое может занять Катя.

Следуйте формату вывода, указанному в примере, как можно точнее.

Example

beautiful.in	beautiful.out
2 3 1 2 0 0 0 2 3 1 0 0	Case #1: Katya's place can be 2.

Problem B. Шахматы (Division 1 Only!)

Input file:	checkmate.in
Output file:	checkmate.out
Time limit:	2 секунды
Memory limit:	256 мебибайт

Дана тороидальная трёхмерная шахматная доска $n \times n \times n$ ($2 \le n \le 4$).

Каждая клетка доски имеет координаты (x, y, z) ($0 \le x, y, z < n$). С клеткой (x, y, z) соседними по стороне являются $((x \pm 1) \bmod n,\ y,\ z)$, $(x,\ (y \pm 1) \bmod n,\ z)$ и $(x,\ y,\ (z \pm 1) \bmod n)$. Различные клетки (x_1, y_1, z_1) и (x_2, y_2, z_2) являются *соседями по углу*, если и минимум, и максимум из «расстояний по модулю» по x, по y и по z равны 1. «Расстоянием по модулю» между числами a и b мы здесь называем $\min(|a - b|,\ n - |a - b|)$.

На тороидальной трёхмерной шахматной доске живут k ($0 \le k \le 4$) агрессивных чёрных королей, которые хотят взять одного не агрессивного белого короля. Белые и чёрные ходят по очереди. За один ход нужно одного короля своего цвета переместить на соседнюю по углу клетку (ход делать обязательно, за один ход перемещается только один король). Короли могут брать друг друга. Для этого нужно походить на клетку с тем, кого король хочет взять. Брать можно только королей другого цвета. Взятая фигура больше не существует (не может ходить и брать другие фигуры, а также не занимает места на поле). Две фигуры не могут стоять на одной и той же клетке одновременно.

Вам даны исходные координаты всех фигур. Первым ходит не агрессивный белый король. Вопрос: смогут ли чёрные короли его взять? Если смогут, то за какое минимальное число ходов (белый король будет сопротивляться и момент своего взятия максимально оттягивать)?

Input

Файл содержит один или несколько тестов.

Каждый тест начинается с чисел n ($2 \le n \le 4$, размер шахматной доски) и k ($0 \le k \le 4$, количество агрессивных королей). Далее идут $(k+1) \cdot 3$ чисел от 0 до $n-1$ — координаты $k+1$ фигуры (сперва 1 белый король, потом k чёрных). Изначально все фигуры стоят на разных клетках доски.

Файл завершается тестом с $n = k = 0$. Этот тест обрабатывать не нужно. Всего тестов не более 10^4.

За каждым числом во входных данных следует непустая последовательность пробелов и/или символов перевода строки.

Output

Для каждого теста выведите «YES», если чёрные короли смогут взять белого, или «NO» в противном случае. Если ответ — «YES», также выведите минимальное число ходов, необходимое, чтобы взять белого короля при оптимальной игре обеих сторон (считать нужно только ходы чёрной стороны).

Пример выходных данных показывает, в каком формате нужно выводить данные. Следуйте этому формату максимально точно.

Example

checkmate.in	checkmate.out
2 4 0 0 0 0 0 1 0 1 0 1 0 0 1 1 1 4 4 2 2 2 0 0 0 0 0 1 0 1 0 0 1 1 0 0	Case #1: YES 1 Case #2: YES 3

Problem C. Codeforces

Input file:	`codeforces.in`
Output file:	`codeforces.out`
Time limit:	2 секунды
Memory limit:	256 мебибайт

Вася любит принимать участие в соревнованиях Codeforces. В каждом соревновании, участникам дается набор из не более, чем пяти задач. За верное решение каждой из них участники получают очки, зависящие от времени, потраченного на решение задачи. Есть и другие способы получить и потерять баллы. Не стоит и говорить, что они могут быть критичны для результатов соревнования.

К примеру, если все задачи очень сложны, победитель может и не решить ни одной задачи. А когда все задачи просты, даже последний в таблице результатов может сдать правильные решения по всем задачам. Вася считает такие наборы задач плохими.

Чтобы формализовать рассуждения, он придумал формулу, определяющую неприятность соревнования. Он считает, что все участники получили различное число очков, так что никакие два не заняли одно и то же место.

Пусть n — число участников. Вася определяет неприятность соревнования как

$$B = \frac{2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f(i,j)^2}{n(n-1)}.$$

Здесь, $f(i,j)$ равно нулю, если участник на i-й позиции решил больше задач, чем участник на j-й позиции. Иначе, $f(i,j)$ равно разнице между числом задач, решенным i-м участником и числом задач, решенным j-м участником.

Input

Ввод состоит из одного или более тестов.

Каждый тест начинается строкой, содержащей целое число n ($2 \le n \le 100\,000$). Во второй строке записаны n целых чисел — количество задач, решенных участниками (начиная с первого в финальной таблице и заканчивая последним; помните, что в соревновании не более пяти задач). Ввод будет завершен тестом с $n = 0$, который не требуется обрабатывать.

Сумма n по всем тестам во вводе не превысит $100\,000$.

Output

Для каждого теста выведите единственную строку с неприятностью соревнования. Ответ может быть целым или рациональным. Во втором случае числитель и знаменатель должны быть взаимно простыми, а знаменатель должен быть положительным.

Следуйте формату вывода, указанному в условии, как можно точнее.

Example

codeforces.in
3 3 2 1 2 0 5 4 0 1 0 0 0
codeforces.out
Case #1: The contest badness is 0. Case #2: The contest badness is 25. Case #3: The contest badness is 1/6.

Problem D. Конкатенация (Division 1 Only!)

Input file:	concat.in
Output file:	concat.out
Time limit:	2 секунды
Memory limit:	256 мебибайт

Дана строка S, состоящая из строчных букв латинского алфавита. Рассмотрим строку $T(S)$, представляющую собой конкатенацию всех подстрок S в лексикографическом порядке.

Например, если $S =$ aba, ее подстроки это {a, b, a, ab, ba, aba}, подстроки в лексикографиеском порядке это {a, a, ab, aba, b, ba}, и таким образом, $T(S) =$ aaababababba.

Найдите i-й символ строки $T(S)$.

Input

Ввод состоит из одного или более тестов.

Каждый тест начинается строкой, содержащей натуральное число m, задающее число запросов. Следующая строка содержит строку S ($1 \le |S| \le 5\,000$). Следующая строка содержит m целых чисел a_i ($1 \le a_i \le |T(S)|$), задающих запросы.

Ввод будет завершен тестом с $m = 0$, который не требуется обрабатывать.

Сумма m по всем тестам во вводе не превысит $5\,000$.

Сумма длин всех строк S не превысит $5\,000$.

Output

Для каждого теста выведите строку из m символов: ответы на запросы. Следуйте формату вывода, указанному в примере, как можно точнее.

Example

concat.in	concat.out
10 aba 1 2 3 4 5 6 7 8 9 10 1 x 1 0	Case #1: aaababababba Case #2: x

Problem E. Словарь (Division 1 Only!)

Input file:	dictionary.in
Output file:	dictionary.out
Time limit:	2 секунды
Memory limit:	256 мебибайт

. . . Оглавление для словаря. . .

Список самых бесполезных книг

Вы нашли словарь. В нём перечислены в лексикографическом порядке некоторые слова с их иероглифическим переводом. Вы уже отсканировали словарь. К сожалению, программа распознавания проигнорировала все иероглифы. Более того, все пробелы тоже исчезли, в результате чего осталась одна длинная строка из строчных букв латинского алфавита.

Никакого лингвистического интереса полученная строка не представляет, поэтому вам предлагается следующая комбинаторная задача: подсчитать число способов разбить эту строку на одно или больше различных слов так, чтобы эти слова оказались перечисленными в лексикографическом порядке.

Input

Ввод состоит из одного или более тестов.

Каждый тест представляет собой строку из n строчных букв латинского алфавита ($1 \le n \le 3\,000$).

Ввод будет завершён строкой, содержащей единственный символ тире («-»), который не требуется обрабатывать.

Сумма n по всем тестам во вводе не превысит $3\,000$.

Output

Для каждого теста выведите число способов разбить строку. Это число может быть достаточно большим, поэтому следует выводить его остаток по модулю $10^9 + 9$.

Следуйте формату вывода, указанному в примере, как можно точнее.

Example

dictionary.in	dictionary.out
a aa ab ba abacaba abracadabra -	Case #1: There are 1 ways. Case #2: There are 1 ways. Case #3: There are 2 ways. Case #4: There are 1 ways. Case #5: There are 7 ways. Case #6: There are 34 ways.

Problem F. Друзья друзей

Input file:	`ffriends.in`
Output file:	`ffriends.out`
Time limit:	2 секунды
Memory limit:	256 мебибайт

Социальные сети изменили значение слова «друг». Когда-то оно подразумевало открытого для вас человека, разделяющего ваши интересы, идеи, что угодно. Теперь это всего лишь запись в базе данных, несколько байт в настройках приватности.

Но суть не в этом. Вы тестируете новый «движок» для социальной сети. Чтобы воспроизвести некую ошибку, вам необходимо найти пару человек, у которых нет общих друзей.

Input

Входные данные состоят из одного или более тестов.

Каждый тест начинается с целого числа n ($1 \le n \le 1\,000$), которое определяет число пользователей социальной сети.

Следующие n строк содержат закодированный граф отношения «дружбы». Строка с номером i описывает i бит: j-й установлен в 1, если пользователи i и j являются «друзьями».

Граф закодирован по одной разновидности схемы *Base64*. Подробнее алгоритм кодирования описан далее.

Биты группируются по шесть штук. Каждая группа из шести бит рассматривается как двоичное число от 0 до 63. Числа от 0 до 25 кодируются буквами от "`A`" до "`Z`", числа от 26 до 51 кодируются буквами от "`a`" до "`z`", числа от 52 до 61 кодируются цифрами от "`0`" до "`9`", числа 62 и 63 кодируются символами "`+`" и "`/`", соответственно. К каждой строке, число бит в которой не кратно шести, добавляются в конец нулевые биты до кратности. Каждая строка кодируется отдельно. Первый символ строки соответствует первым шести битам, второй — второй группе из шести битов, и так далее.

Входные данные завершаются тестом, где $n = 0$, который не нужно обрабатывать.

Сумма n по всем тестам во входных данных не превысит $1\,000$.

Output

Для каждого теста выведите пару порядковых номеров пользователей, не имеющих общих «друзей», либо сообщение об ошибке, если такой пары нет. Если решений несколько, выведите пару с наименьшим возможным первым числом. Если таких пар также несколько, выберите пару с наименьшим вторым числом.

Следуйте формату вывода, указанному в примере, как можно точнее.

Example

ffriends.in
3 A g Q 3 A g w 0
ffriends.out
Case #1: Members 1 and 2 have no common friends. Case #2: Social graph is too dense.

Problem G. Подготовка к игре (Division 1 Only!)

Input file:	`game.in`
Output file:	`game.out`
Time limit:	2 секунды
Memory limit:	256 мебибайт

> «Трое играющих берут четыре фишки, причём пятый игрок все время выкидывает. После того, как лиса оказывается съеденной, она делает четыре хода назад.»
>
> кинофильм «Подкидыш»

А теперь пора подготовить игровое поле. На нём n позиций, некоторые из которых соединены с другими. Соединения устроены так, что существует ровно один путь между каждой парой позиций. Также есть несколько фишек, которые можно расставлять на позициях. Есть лишь одно ограничение: никакие две фишки не могут быть расположены на расстоянии меньшем, чем c ходов.

Чтобы выбрать действительно случайное расположение, найдите для начала число способов расставить фишки. Разрешается расставлять любое число фишек. Или не расставлять их вовсе.

В чём же цель игры, спросите вы? Я не знаю. Может быть, вам и предстоит изобрести её.

Input

Входные данные состоят из одного или более тестов.

Каждый тест начинается строкой, в которой записаны целое число n ($1 \le n \le 10\,000$), которое определяет число позиций на поле, и целое число c ($1 \le c \le \min(n, 500)$), которое определяет минимальное расстояние между фишками. Следующая $n-1$ строка описывает пары полей, которые соединены между собой.

Входные данные завершаются тестом, где $n = c = 0$, который не нужно обрабатывать.

Сумма n по всем тестам во входных данных не превысит $10\,000$.

Output

Для каждого теста выведите искомое число способов. Поскольку это число может быть очень большим, его следует вывести по модулю 10^6.

Следуйте формату вывода, указанному в примере, как можно точнее.

Example

game.in	game.out
5 2 1 2 1 3 1 4 4 5 2 2 1 2 2 1 1 2 0 0	Case #1: 14 Case #2: 3 Case #3: 4

Problem H. Пересечение списков

Input file:	lists.in
Output file:	lists.out
Time limit:	2 секунды *(3 секунды для Java)*
Memory limit:	256 мебибайт

НИИ Данны Строк приготовил очередное задание для Васи. Дано n списко возрастающих целых чисел. Требуется написать программу, которая определит количество элементов в пересечении этих списков, то есть число элементов, присутствующих в каждом из них.

Вася знает, что его программу будут проверять на списках, сгенерированных определенным образом. Каждый список содержит ровно n целых чисел (y_0, y_1, ..., y_{n-1}) и зависит от пяти параметров (x_0, y_0, a, b, m). Элементыs y_1, y_2, ... вместе с дополнительными значениями x_1, x_2, ... генерируются следующим образом:

$$\begin{aligned} x_i &= (a \cdot x_{i-1} + b) \bmod 4\,294\,967\,296, \\ y_i &= y_{i-1} + 1 + (x_i \gg m). \end{aligned}$$

Здесь $u \gg v$ означает сдвиг двоичного представления u на v бит вправо. Это фактически то же, что и целочисленное деление u на 2^v. Если m превышает 31, значение $x_i \gg m$ считается равным нулю.

Input

Ввод состоит из одного или более тестов. Каждый тест начинается строкой с единственным целым числом n ($1 \le 8\,000$), определяющим количество списков. Каждая из следующих n строк содержит пять целых чисел x_0, y_0, a, b, m ($0 \le x_0, y_0, a, b, m < 4\,294\,967\,296$), задающих параметры списков. Ввод будет завершен тестов с $n = 0$, который не требуется обрабатывать.

Сумма n по всем тестам во вводе не превысит $8\,000$.

Output

Для каждого теста выведите строку с размером искомого пересечения. Следуйте формату вывода, указанному в примере, как можно точнее.

Example

lists.in
3 0 0 1 1 0 1 0 30 239 1000000 0 0 0 2 1 2 0 0 2 5 1 1 1 5 2 1 0

lists.out
Case #1: Intersection contains 2 integer(s). Case #2: Intersection contains 0 integer(s).

Problem I. Прямоугольники (Division 1 Only!)

Input file:	`rectangles.in`
Output file:	`rectangles.out`
Time limit:	6 секунд
Memory limit:	256 мебибайт

Вам дана квадратная матрица целых чисел. У каждой ячейки матрицы так же есть ширина и длина, равные единице.

Вам нужно найти такую подматрицу, что величина $\frac{S}{P}$ принимает масимально возможное значение. Здесь S — сумма всех чисел в подматрице, а P — длина периметра подпрямоугольника, соответствующего подматрице.

Input

Ввод состоит из одного или более тестов.

Каждый тест начинается со стоки, содержащей одно целое число n ($1 \le n \le 300$), обозначающее размер матрицы. Следующие n строк содержат по n целых чисел: значения элементов матрицы. Все элементы матрицы не превосходят 10^9 по абсолютной величине.

Ввод завершается тестом с $n = 0$, который не требуется обрабатывать.

Сумма чисел n по всем тестам не превосходит 300.

Output

Для каждого теста, выведите значение функции $\frac{S}{P}$. Число нужно выводить с максимально возможной точностью. Значение, которое вы выведете, должно отличаться от правильного не более чем на 10^{-5}.

Затем выведите x_1, y_1 и x_2, y_2 — координаты углов подматрицы. Первый из них должен быть верхним-левым, а второй нижним-правым.

Следуйте формату вывода, указанному в примере, как можно точнее.

Example

rectangles.in
2 100 100 1 1 0
rectangles.out
Case #1: The maximal value is 33.333333333, rectangle corners are (1, 1) and (2, 1).

Problem J. Последовательность Вилсона

Input file:	wilson.in
Output file:	wilson.out
Time limit:	2 секунд
Memory limit:	256 мебибайт

Для данного n, Вася должен опредеить n-й элемент последовательности Вилсона:

$$w_n = (n-1)! \bmod n.$$

Input

Ввод состоит из одного или более тестов. Всего тестов не более 10 000.

Каждый тест представляет собой строку с единственным целым числом n ($1 \le n \le 1\,000\,000$). Ввод будет завершен тестом с $n = 0$, который не требуется обрабатывать.

Output

Для каждого теста выведите единственную строку с соответствующим элементом последовательности. Следуйте формату вывода, указанному в примере, как можно точнее.

Example

wilson.in
1 2 3 5 6 21 0

wilson.out
Case #1: 1st Wilson number is equal to 0. Case #2: 2nd Wilson number is equal to 1. Case #3: 3rd Wilson number is equal to 2. Case #4: 5th Wilson number is equal to 4. Case #5: 6th Wilson number is equal to 0. Case #6: 21st Wilson number is equal to 0.

Problem K. Division (Division 2 Only!)

Input file:	`division.in`
Output file:	`division.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Для заданного основания системы счисления b, числителя x и знаменателя y (числитель и знаменатель заданы в b-ичной системе счисления) найдите длину периода дроби $q = x/y$ в b-ичной системе счисления.

Напомним, что цифры в записи числа q в b-ичной системе счисления определяются следующим образом:

$$q = q_n q_{n-1} \dots q_0 . q_{-1} q_{-2} \dots = \sum_{i=-\infty}^{n} q_i b^i$$

К примеру, $1_{10}/4_{10} = 0.25_{10}$, $1_4/10_4 = 0.1_4$, $3_{10}/11_{10} = 0.272727\ldots_{10} = 0.(27)_{10}$.

В первых двух случаях получившаяся дробь конечна, все последующие цифры равны нулю; в этом случае мы считаем длину периода равной нулю (вариант с записью типа $0.24(9)_{10}$ не проходит, так как нам нужно вычислить минимальную длину периода). В третьем примере длина периода равна двум.

Input

Первая строка входного файлс содержит целое положительное число T ($1 \le T \le 250$) — количество дробей. В последующих T строках задаются дроби. Каждая дробь задаётся целым числом b в десятичной записи — основанием системы счисления, далее следуют два целых числа в b-ичной записи — числитель x и знаменатель y соответственно ($0 < y \le 10^5_{10}$, $0 \le x \le y$). Цифры, большие, чем 9, представляются буквами от '`a`' до '`z`' соответственно (при этом регистр не имеет значения, то есть записи, к примеру, $1a$ и $1A$ обозначают одно и то же число).

Output

Для каждой дроби выведите в отдельной строке одно число, записанное в десятичной системе счисления — длину периода данной дроби в b-ичной записи.

Example

division.in	division.out
4 10 1 4 4 1 10 10 4 11 36 Ic Pc	0 0 2 9

Problem L. Insects (Division 2 Only!)

Input file:	`insects.in`
Output file:	`insects.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

По длинной деревянной рейке ползают несколько насекомых. Их движение описывается следующим образом: каждое насекомое ползёт в одном из двух направлений (слева направо или справа налево) с постоянной скоростью 1 cm/sec. Как только два насекомых встречаются в одной точке, каждое из них меняет направление на противиположное и продолжает движение. Если насекомое доползло до конца рейки, оно сваливается вниз и в дальнейшем при расчётах взаимодействий с другими насекомыми не учитывается.

Ваша задача — промоделировать движение насекомых. В рамках данной задачи размером каждого из насекомых пренебречь.

Input

В первой строке входного файла заданы два целых числа L и A. L — длина рейки в сантиметрах ($1 \le L < 10^5$), A — количество насекомых на рейке в начале движения ($1 \le A \le L + 1$). Каждая из последующих A строк содержит целое неотрицательное число X_i — начальную координату i-го насекомого, после которого через пробел задана одна из двух букв: "`L`", если насекомое движется справа налево (то есть в направлении нуля) или "`R`", если насекомое движется слева направо (то есть от нуля). Ни для каких двух различных насекомых начальные координаты не совпадают.

Output

Выведите одно число — точное время, когда последнее насекомое (или два, если они сделают это одновременно) дойдёт до конца рейки. Если возможно, выведите целое число, в противном случае выведите ровно один знак после десятичной точки.

Examples

insects.in	insects.out
98765 1 0 R	98765
12 2 0 L 12 R	0
19 6 8 L 7 L 12 L 18 R 3 R 9 L	16

Problem M. Sentences (Division 2 Only!)

Input file:	sentences.in
Output file:	sentences.out
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Эксперименты показали, что человек может понять напечатанное с перестановкой букв слово, если соблюдается следующее правило: первая и последняя буква слова остаются неизменными. При этом буквы посередине, как утверждается, могут быть переставлены как угодно.

Однако подобное утверждение верно далеко не всегда: в некоторых случаях одной такой записи соответствуют несколько слов: например, записи "colud" могут соответствовать английские слова "cloud" или "could".

Вам задан словарь, в котором перечислены допустимыес слова, а также несколько предложений (слов, разделённых пробелами), каждое из слов в котором модифицировано описанным выше способом. Для каждого из заданных предложений установите, сколько исходных предложений, составленных из словарных слов, может ему сооответствовать.

Input

В первой строке входного файла задано количество слов в словаре — целое число n ($0 \le n \le 10^4$). Последующие n строк содержат эти слова по одному на строку. Каждое слово имеет длину не более 100 символов и состоит из строчных и прописных латинских букв. Далее следует строка, содержащее одно число — количество предложений. Каждая из последующих m строк содержит по одному предложению. Предложения состоят из строчных и прописных английских букв, а также пробелов. Длина предложения не превышает 10^4 символов.

Output

Для каждого предложения, указанного во входном файле, выведите в отдельной строке количество различных исходных предложений, составленных из слов заданного словаря, из которых можно получить заданное описанной в условии перестановкой букв.

Example

sentences.in	sentences.out
5 xyxyx xxyyx xyzxx Xxzyx xx 4 xyxyx xyzzx xxyzx xyxzx xyyxx xyxyx	2 0 1 4

Problem N. Transportation (Division 2 Only!)

Input file:	`transport.in`
Output file:	`transport.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Вам задан план города, представляющий из себя несколько перекрёстков, соединённых улицами с двусторонним движением. Для каждой улицы известен максимальный вес, который можно перевезти по этой улице. Перекрёстки занумерованы от 1 (вокзал) до n (морской порт). Ваша задача — вычислить, какой максимальный вес можно перевезти по городским улицам от вокзала до морского порта. Гарантируется, что путь от вокзала до морского порта существует.

Input

В первой строке входного файла заданы два целых числа n ($1 \le n \le 1000$) — количество перекрёстков и m — количество соединяющих их улиц (два перекрёстка не может соединять напрямую более одной улицы). В последующих m строках заданы улицы. Каждая улица задана тремя целыми числами — номерами a и b ($a \ne b$, $1 \le a, b \le n$) перекрёстков, которые она соединяет, и максимальным разрешённым весом w ($1 \le w \le 10^6$).

Output

Выведите максимальный вес, который можно перевезти за одну поездку от вокзала до морского порта.

Example

transport.in	transport.out
3 3 2 1 2 3 1 4 2 3 6	4

Problem O. Travel (Division 2 Only!)

Input file:	travel.in
Output file:	travel.out
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Во время путешествий Вася никогда не записывает свой маршрут целиком. Зато он сохраняет все билеты. На каждом билете написано, от какой станции до какой ехал Вася.

Через какое-то время после завершения своего путешествия Вася хочет восстановить маршрут в виде последовательности посещённых им станций в правильном порядке. Помогите ему в этом.

Input

В первой строке входного файла содержится целое число T ($3 \le T \le 340$) — количество билетов у Васи. Каждая из последующих $T-1$ строк описывает один билет и содержит два названия станций, разделённых пробелом: станцию отправления и станцию прибытия соответственно. Названия станций состоят не более, чем 30, заглавных и строчных латинских букв.

Output

Выведите T строк, по одной станции на строку, в порядке, в котором Вася проехал эти станции во время путешествия.

Example

travel.in	travel.out
4 Petushki NizhnyNovgorod Moscow Petushki Ikstlan Moscow	Ikstlan Moscow Petushki NizhnyNovgorod