

Problem A. Бомбардировка

Input file: bomb.in
Output file: bomb.out
Time limit: 2 seconds
Memory limit: 256 megabytes

В этой задаче вам необходимо промоделировать поведение отряда на поле боя в условиях бомбардировки. Будем считать, что поле боя является бесконечной плоскостью и в каждый целочисленный момент времени $t \geq 0$ отряд находится в некоторой точке (x, y) этой плоскости с целочисленными координатами. В момент времени $t_0 = 0$ отряд находится в точке $(0, 0)$. Если в момент времени t отряд находится в точке (x, y) , то в момент времени $t + 1$ он может оказаться в одной из точек $(x + dx, y + dy)$, где $|dx|, |dy| \leq 1$. Другими словами, отряд может остаться на месте или переместиться в одну из соседних точек в 8-ми базовых направлениях.

Известно, что на поле будет сброшено N бомб. Каждая i -я бомба, $1 \leq i \leq N$, характеризуется величинами X_i, Y_i, R_i, t_i , где t_i — момент времени, в который состоится взрыв, (X_i, Y_i) — точка плоскости, в которую будет сброшена бомба, и R_i — мощность бомбы. Отряд погибает от взрыва i -й бомбы в том и только том случае, если в момент времени t_i он находится в такой точке (x, y) , что $|x - X_i| \leq R_i$ и $|y - Y_i| \leq R_i$.

Определите количество различных точек, в которых отряд может оказаться в заданный момент времени T , не погибнув.

Input

Входной файл содержит описание одного или нескольких наборов входных данных. В первой строке записано общее количество наборов M ($1 \leq M \leq 1,000$). Далее следует описание M наборов.

Описание каждого набора начинается со строки, содержащей целые числа N и T ($0 \leq N \leq 100$, $2 \leq T \leq 1\,000\,000\,000$).

Каждая из следующих N строк описывает одну бомбу и содержит целые числа X_i, Y_i, R_i и t_i ($-1\,000\,000\,000 \leq X_i, Y_i \leq 1\,000\,000\,000$, $0 \leq R_i \leq 1\,000\,000\,000$, $0 \leq t_i \leq T$). Бомбы записаны в файле в порядке их падения, но при этом возможно одновременное падение нескольких бомб, т.е. $t_1 \leq t_2 \leq \dots \leq t_N$.

В случае, если во входном файле записано более одного набора данных, для каждого из них будет выполнено $N \leq 10$.

Output

Для каждого набора входных данных необходимо вывести в отдельной строке количество различных точек, в которых отряд может оказаться в момент времени T , не погибнув. Если гибель отряда до момента времени T неизбежна, выведите 0.

Example

bomb.in	bomb.out
3	20
1 2	40401
2 0 1 1	0
0 100	
1 100	
0 0 50 50	

Problem B. Раскраска графа

Input file: coloring.in
Output file: coloring.out
Time limit: 2 seconds
Memory limit: 256 megabytes

Дан неориентированный граф $G = (V, E)$ без петель и кратных ребер. Пусть $u \in V$, тогда $V_u = \{x \mid (u, x) \in E\}$, $E_u = \{(x, y) \mid (x, y) \in E, x \in V_u, y \in V_u\}$. Граф G называется локально связным, если для любой вершины u граф $G_u = (V_u, E_u)$ содержит не более одной компоненты связности.

Известно, что вершины графа G можно раскрасить в три цвета так, чтобы никакие две смежные вершины не были покрашены в один цвет. Требуется построить любую такую раскраску.

Input

В первой строке файла записаны два числа N и M ($1 \leq N \leq 100000$, $0 \leq M \leq 200000$). Далее в M строках по два числа a, b ($1 \leq a < b \leq N$) задают ребро между вершинами a и b .

Output

Единственная строка выходного файла должна содержать N символов '1', '2', '3', где i -й символ соответствует цвету, в который необходимо покрасить вершину i . Если существует несколько раскрасок, выведите любую из них.

Example

coloring.in	coloring.out
3 3 1 2 2 3 1 3	123
4 5 1 2 1 3 1 4 2 3 2 4	1233

Problem C. Турнир по переписке

Input file: `contest.in`
Output file: `contest.out`
Time limit: 2 seconds
Memory limit: 256 megabytes

Не в столь уж далёкие времена, когда быстрого интернета ещё не было, шахматисты проводили турниры по переписке. При организации турнира определялись N участников, которые выбирали из своих рядов старосту. Игра в турнире проходит так: в каждой партии игрок пишет письмо сопернику с указанием очередного хода и ждёт от него письма с ответным ходом. Если партия завершена, об этом информируется староста турнира.

На обдумывание хода даётся до недели... Если учесть время на пересылку писем, то турниры могут длиться по году и более. Естественно, в это время участники по различным причинам могут отказаться от дальнейшего участия в турнире. Об отказе участник также информирует старосту турнира. Отказавшемуся участнику засчитываются поражения во всех партиях, которые он не успел завершить. Если партия не сыграна из-за отказа от участия обоих игроков, им засчитывается обоюдное поражение.

Обычно турнир завершается после того, как все партии сыграны... Однако староста с согласия оставшихся игроков может принудительно завершить затянувшийся турнир. При этом все партии между участниками, оставшимися в турнире, информация о которых не поступила старосте, считаются завершившимися вничью.

Одной из систем проведения турниров является однокруговая система. При проведении турнира по такой системе каждый участник должен сыграть ровно одну партию с каждым из остальных участников. Староста одного из турниров по переписке, проводящегося по однокруговой системе, решил завершить его принудительно и поручил Вам составить итоговую турнирную таблицу на основании M сообщений, полученных им. Сделайте это!

Input

Первая строка содержит значения N и M ($2 \leq N \leq 100000$, $0 \leq M \leq 100000$). Далее следуют M строк с информацией о ходе турнира. Возможны два типа строк:

- Строка с информацией об окончании партии. Она содержит три числа i , j , k , разделённые пробелами: $1 \leq i < j \leq N$ — номера игроков, сыгравших партию, k — результат партии (1 — выиграл игрок i , -1 — выиграл игрок j , 0 — партия закончилась вничью). Для каждой пары (i, j) может быть не более одной строки такого типа.
- Строка с информацией об отказе участника от дальнейших игр. Она содержит номер участника и ноль. Эти числа записаны через один или несколько пробелов. Для каждого участника может быть не более одной строки такого типа.

Порядок строк не обязательно соответствует порядку поступления информации, так что после строки об отказе какого-либо участника от игры может следовать информация о сыгранных им партиях.

Output

Выходной файл должен содержать N строк. Каждая строка этого файла соответствуют одному участнику и должна содержать его номер и количество набранных этим участником очков. За победу присуждается одно очко, за ничью — 0,5 очка, за поражение очки не присуждаются. Строки

должны быть упорядочены по убыванию набранных очков, а в случае их равенства — по возрастанию номера участника.

Example

contest.in	contest.out
3 3 1 2 1 2 3 0 1 0	3 1.5 1 1.0 2 0.5
7 2 7 0 6 0	1 4 2 4 3 4 4 4 5 4 6 0 7 0

Problem D. Дуэль

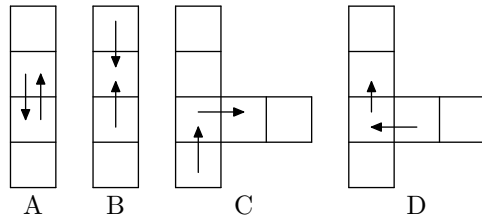
Input file: `duel.in`
Output file: `duel.out`
Time limit: 2 seconds
Memory limit: 256 megabytes

Два друга О. и Л. неожиданно поссорились, и сегодня между ними должна состояться дуэль. Сейчас они оба отдают себе отчет в том, что вчера погорячились, но гордость не позволяет никому из них отказаться от поединка. Секунданты, также не желающие проведения поединка, предложили необычную схему проведения дуэли.

Местом дуэли выбран парк с одной главной аллеей длиной L метров и отходящими от нее K боковыми тупиковыми аллеями. Все боковые аллеи не пересекаются друг с другом, а точки их соединения с главной аллеей отстоят друг от друга на положительное целое число метров. В начале поединка дуэлянты стоят перед входом в главную аллею и по сигналу секундантов начинают одновременно сходить к себе с скоростью 1 м/с. Во время движения дуэлянты могут заходить в боковые аллеи (не более одного раза в каждую), проходить по ним до конца, а потом возвращаться обратно на главную аллею. Изменять направление движения по главной аллее не разрешается.

Дуэль начинается тогда, когда противники сближаются на расстояние, меньшее 1 метра. Если же оба дуэлянта пройдут по главной аллее от одного конца до другого и не встретятся друг с другом, дуэль считается состоявшейся и закончившейся вничью (такой исход будет, наверное, наилучшим).

Поясним принципы начала дуэли. Пусть план парка разбит на клетки размером 1×1 метр. Тогда противники за одну секунду перемещаются в соседние клетки. В ситуации А и В дуэль неизбежна, а в ситуациях С и D дуэлянты успеют разойтись, хотя и находятся в соседних клетках.



Определите, может ли дуэль завершиться вничью, если оба противника будут стремиться к этому?

Input

Первая строка содержит значения L и K ($2 \leq L \leq 10000$, $0 \leq K \leq 20$, $K \leq L$). Далее следуют K строк с описаниями боковых аллей. Каждая из этих строк содержит два числа — расстояние C_i от того конца главной аллеи, где в начале стоит О., до начала описываемой боковой аллеи, и ее длину P_i ($2 \leq C_i \leq L - 1$, $1 \leq P_i \leq 10000$). Все числа в строках разделяются единственным пробелом.

Output

Первая строка этого файла должна содержать значение “Yes” или “No” (без кавычек) в зависимости от того, может ли дуэль завершиться вничью. В случае положительного ответа в выходной файл записывается информация об одном из маршрутов дуэлянтов, которые приведут к такому исходу (во второй строке — информация о маршруте О., в третьей — о маршруте Л.). Каждая из этих строк содержит информацию о числе боковых аллей, в которые зашел дуэлянт, и номера таких аллей в порядке возрастания. Аллеи нумеруются, начиная с единицы, в соответствии с порядком их описаний во входном файле.

Example

duel.in	duel.out
11 2 4 1 6 3	Yes 1 1 1 2
11 2 4 1 10 3	No

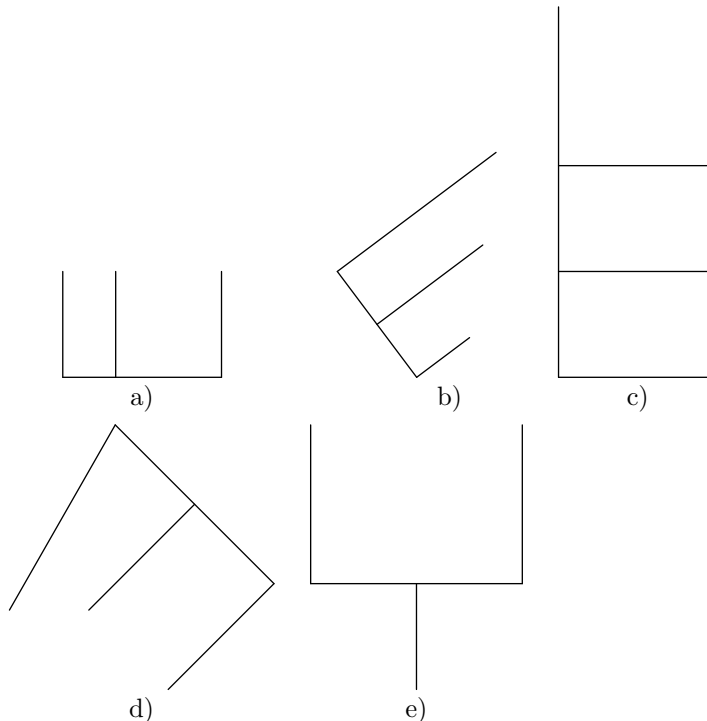
Problem E. Буква Ш

Input file: sh_shape.in
Output file: sh_shape.out
Time limit: 2 seconds
Memory limit: 256 megabytes

Будем говорить, что четыре отрезка ненулевой длины (“основание” и три “палочки”) образуют букву Ш, если:

1. “палочки” лежат на трех различных параллельных друг другу прямых;
2. прямые, на которых лежат “палочки”, перпендикулярны прямой, на которой лежит “основание”;
3. одни из концов каждой “палочки” лежит на “основании”, причём концы двух “палочек” совпадают с различными концами “основания”;
4. другие стороны “палочек” лежат по одну сторону от “основания”.

Таким образом, отрезки на рис. а) – б) образуют букву Ш, а отрезки на рис. в) – е) — не образуют.



На плоскости заданы N отрезков ненулевой длины, пронумерованных, начиная с единицы. Определите, сколько различных четвёрок этих отрезков образуют букву Ш. Четвёрки отрезков различаются, если в них входят отрезки с различными номерами.

Input

Первая строка содержит величину N ($1 \leq N \leq 100000$). Далее следуют N строк с координатами концов каждого отрезка в формате (x_1, y_1, x_2, y_2) . Все координаты — целые числа, не превосходящие по модулю 10^9 .

Output

Искомое количество различных четверок отрезков.

Example

sh_shape.in	sh_shape.out
10 0 0 9 3 -1 3 0 0 3 1 2 4 5 5 7 -1 7 -1 7 2 6 -1 6 2 9 -1 9 0 9 3 8 6 6 2 5 5 9 -1 6 -1	3

Примечание: букву Ш образуют четвёрки отрезков (1, 2, 3, 8), (1, 2, 8, 9), (5, 6, 7, 10).

Problem F. Метро

Input file: metro.in
Output file: metro.out
Time limit: 2 seconds
Memory limit: 256 megabytes

Система метро города Энска состоит из N линий и M станций. Каждая линия не пересекается сама с собой. Это означает, что при движении от одной конечной станции до другой поезд метро проходит по каждой промежуточной станции один раз. Кольцевых линий метро нет.

Все станции имеют уникальные названия. На некоторых станциях возможна пересадка с одних линий на другие. Точнее, если станция S присутствует в списке станций для линий A и B , то на этой станции можно пересесть с линии A на B и обратно. Пересадка с одной линии на другую может быть выполнена на различных станциях.

Стоимость пользования метро складывается из двух частей:

- Стоимости проезда, которая зависит от количества перегонов между соседними станциями, которые проехал пассажир. Разные линии метро управляются разными компаниями, поэтому стоимость проезда одного перегона может быть различной для разных линий.
- Стоимости пересадок. Стоимость одной пересадки устанавливается муниципалитетом и одинакова для всей системы метро.

Вам необходимо проехать от станции S_1 к станции S_2 . Определите, минимальную стоимость поездки.

Input

Первая строка содержит значения M и N ($2 \leq M \leq 100$, $1 \leq N \leq 100$). Во второй строке записаны целые величины c_0, c_1, \dots, c_N — стоимость пересадки и стоимость проезда одного перегона на каждой линии ($0 \leq c_0 \leq 1000$, $1 \leq c_i \leq 1000$, $i = \overline{1, N}$). Третья строка содержит величины K_i ($i = \overline{1, N}$) — количество станций на каждой линии. Далее следует N описаний каждой линии. Описание i -й линии состоит из K_i строк и содержит названия станций, через которые проходит поезд метро этой линии в порядке, соответствующему маршруту его движения от одной конечной станции до другой. Предполагается, что обратный рейс проходит по этим же станциям в обратном порядке. Длина названия станции не превосходит 100 символов. Наконец, последние две строки содержат названия станций S_1 и S_2 . Название станции состоит из букв латинского алфавита, цифр и пробелов, причем не может начинаться или заканчиваться пробелом.

Output

Единственная строка должна содержать искомую стоимость поездки. Если станции S_1 и S_2 совпадают, результат должен быть равен нулю. Если поездка невозможна по каким-либо причинам, выведите в выходной файл значение -1 .

Example

metro.in	metro.out
4 2 5 3 8 3 2 Station1 Station2 Station3 Station2 Station4 Station3 Station4	16
9 3 5 10 50 20 4 4 5 Hotel de Ville Central Park Banco Nacional Pirate Bay Gendarmerie Banco Nacional Hauptbahnhof Hotel de Ville Levee Central Park Hauptbahnhof Cathedral Airport Airport Hotel de Ville	75
9 3 5 10 50 20 4 4 5 Hotel de Ville Central Park Banco Nacional Pirate Bay Gendarmerie Banco Nacional Hauptbahnhof Hotel de Ville Levee Central Park Hauptbahnhof Cathedral Airport Hauptbahnhof Gendarmerie	90

Problem G. Галлеоны, сикли и кнаты

Input file: money.in
Output file: money.out
Time limit: 2 seconds
Memory limit: 256 megabytes

Оказывается, маги тоже пользуются деньгами! У них в ходу золотые, серебряные и бронзовые монеты, называемые соответственно галлеонами, сиклями и кнатами. Один галлеон содержит семнадцать сиклей, а один сикль — двадцать девять кнатов. Все денежные данные (цены, наличные деньги) формулируются в галлеонах, сиклях и кнатах, причём количество сиклей не превышает 16, а количество кнатов — 28.

Гарри Поттер перед поступлением в Хогвардс получил в банке «Гринготтс» на обзаведение определённую сумму. Однако поступающим надо купить абсолютно необходимые для учёбы вещи: волшебную палочку, сову, мантии, оловянный котёл, медные весы...

Сколько денег окажется у Гарри Поттера после всех покупок?

Input

Первая строка содержит количество денег, полученных Гарри в банке «Гринготтс» (три числа — количество галлеонов, сиклей и кнатов, записанные через пробелы). Во второй строке записано число N — количество предметов, которые необходимо купить. Далее следуют N строк с ценами каждого предмета (в том же формате, что и первая строка).

Ограничения на данные: $0 \leq N \leq 100000$, количество галлеонов в каждой из строк также не превосходит 100000.

Output

Единственная строка этого файла содержит информацию о количестве денег, оставшихся у Гарри после совершения всех покупок (в том же формате, что и первая строка входного файла). Если денег не хватит, выведите в выходной файл число -1 .

Example

money.in	money.out
5 16 10 2 3 9 21 0 4 0	2 2 18
1000 0 0 5 200 0 0 200 0 0 200 0 0 200 0 0 200 0 1	-1

Problem H. Простое число

Input file: prime.in
Output file: prime.out
Time limit: 2 seconds
Memory limit: 256 megabytes

По заданному числу N ($1 \leq N \leq 10^7$) определите, является ли число $M = N^4 + 4$ простым.

Input

Первая строка содержит единственное число T ($1 \leq T \leq 10^5$) — количество тестов в файле. Далее каждая из T строк содержит единственное число N . Сумма всех N в файле не превосходит 10^7 .

Output

Для каждого теста, если число M является простым, выведите строку "Prime" (без кавычек). В противном случае выведите наименьший делитель числа M , не равный единице.

Example

prime.in	prime.out
3	Prime
1	2
2	5
3	

Problem I. Дороги

Input file: roads.in
Output file: roads.out
Time limit: 2 seconds
Memory limit: 256 megabytes

В одной стране есть сеть дорог. Всего в стране N ($2 \leq N \leq 20$) городов и M двунаправленных дорог ($1 \leq M \leq 20000$). Из любого города можно проехать в любой другой используя эти дороги (возможно не напрямую). Между парой городов может быть несколько дорог, что помогает местным жителям обходиться без заторов. Однако все дороги находятся в плохом состоянии и требуют ремонта. Правительство хочет отремонтировать $N - 1$ дорожку, а остальные закрыть. Но на дороги, которые будут отремонтированы, естественно накладывается ограничение: по ним должно быть можно проехать из любого города в любой другой.

Правителя интересует средняя стоимость ремонта набора из $N - 1$ дороги (рассматриваются только наборы удовлетворяющие условию связности). Другими словами, если существует T способов выбрать $N - 1$ дорожку для ремонта и стоимости ремонта в каждом из них равны $Cost_1, Cost_2, \dots, Cost_T$ соответственно, то правителя интересует величина $(Cost_1 + Cost_2 + \dots + Cost_T)/T$. Два способа выбора дорог для ремонта считаются различными, если существует дорога, которая входит в один из них, но не входит в другой.

Input

В первой строке входного файла записаны два числа N и M . Далее в M строках идет описание дорог. Дорога описывается тройкой a, b, c ($1 \leq a < b \leq N, 1 \leq c \leq 1000$), где a, b — города, которые дорога соединяет; c — стоимость ремонта.

Output

В единственной строке строке выведите среднюю стоимость ремонта дорог с абсолютной погрешностью не более 10^{-3} .

Example

roads.in	roads.out
2 1 1 2 100	100.0000000
3 4 1 2 100 2 3 100 1 3 200 1 3 100	240.0000000

Problem J. Округление

Input file: round.in
Output file: round.out
Time limit: 2 seconds
Memory limit: 256 megabytes

Учёным удалось получить значение некоей положительной физической величины Q с большой точностью. К сожалению, для большинства практических приложений такая точность не нужна...

Вам требуется выполнить округление величины Q до заданного числа разрядов. Согласно правилам округления, число в десятичной системе округляют до N -ого знака в зависимости от $N + 1$ -ого знака:

- если $N + 1$ -ый знак < 5 , то N -ый знак сохраняют, а $N + 1$ -ый и все последующие обнуляют;
- если $N + 1$ -ый знак ≥ 5 , то N -ый знак увеличивают на единицу, а $N + 1$ -ый и все последующие обнуляют.

Input

Первая строка содержит десятичное значение величины Q . Длина этой строки, а также количество цифр в целой и дробной части не превосходит 200. Разделитель целой и дробной части (точка) обязательно присутствует в этой строке. Целая часть содержит хотя бы одну цифру, однако незначащих нулей в ней нет.

Вторая строка файла содержит число N — количество разрядов, до которых необходимо округлить величину Q (N не превосходит количества разрядов в дробной части величины Q и не менее -250).

Если $N > 0$, Вам необходимо оставить N разрядов в дробной части числа Q . Если $N = 0$, выполните округление до целого. Если $N < 0$, выполните округление так, чтобы $-N$ младших разрядов в целой части числа были равны нулю.

Output

Единственная строка должна содержать округлённую величину Q . Если $N > 0$, дробная часть должна содержать ровно N цифр. Если $N \geq 0$, незначащие нули и десятичную точку выводить не нужно.

Example

round.in	round.out
3.1415926 4	3.1416
2.916787153130130643140 0	3
2.916787153130130643140 -1	0

Problem K. Последовательность

Input file: sequence.in
Output file: sequence.out
Time limit: 2 seconds
Memory limit: 256 megabytes

Задана последовательность из N целых чисел, модуль каждого из которых не превосходит 10^9 . Элементы последовательности перенумерованы, начиная с единицы. Над последовательностью может быть выполнена одна из двух операций:

- Удаление элементов с номерами $a, a + 1, \dots, b$ ($1 \leq a \leq b$, b не превосходит количества элементов в последовательности). После этого происходит перенумерация оставшихся элементов последовательности.
- Поиск k -го по величине элемента последовательности ($1 \leq k$, k не превосходит количества элементов в последовательности).

Имеется M запросов на выполнение этих операций, которые должны быть обработаны друг за другом, так что результаты выполнения предыдущих запросов влияют на последующие. Вам требуется определить результаты операций поиска.

Input

Первая строка этого файла содержит величины N и M ($1 \leq N, M \leq 200000$). Далее следуют N строк с элементами последовательности и M строк с запросами. Запрос на операцию удаления имеет вид “1 a b”, а запрос на поиск — “2 k” (оба запроса записаны без кавычек). Всю цепочку запросов можно выполнить. В цепочке запросов присутствует T запросов на поиск ($1 \leq T \leq M$).

Output

Каждая из T строк должна содержать результат выполнения одного запроса на поиск. Порядок строк соответствует порядку выполнения таких запросов.

Example

sequence.in	sequence.out
5 3 2 91 45 94 1 2 1 2 5 1 1 1	1 94
10 10 9 3 8 1 0 4 5 2 7 6 1 1 1 2 1 1 4 4 2 1 1 1 1 2 1 1 1 1 2 1 1 1 4 2 1	0 1 1 1 6