

Problem A. Beads

Input file: Standard input
Output file: Standard output
Time limit: 10 seconds
Memory limit: 256 mebibytes

На длинную нитку вплотную друг к другу нанизаны n разноцветных шариков одного радиуса. Для того, чтобы разрезать эту нитку на ожерелья, состоящие из k шариков, применяется специальная машина, которая при прохождении каждого k -го шарика разрезает нитку, то есть первое ожерелье содержит шарики с номерами $1, \dots, k$, второе — с номерами $k + 1, \dots, 2k$, и так далее. Если n не делится на k , то последняя часть, содержащая менее k шариков, утилизируется. Обозначим цвета шариков целыми положительными числами.

Требуется выбрать k таким образом, чтобы среди получившихся заготовок было как можно больше различных. Две заготовки считаются одинаковыми, если последовательности цветов шариков в этих заготовках либо одинаковы, либо «перевёрнуты», то есть заготовки, к примеру, $(1, 2, 3)$ и $(3, 2, 1)$ одинаковы.

Например, в случае, если изначально последовательность шариков выглядела как:

$(1, 1, 1, 2, 2, 2, 3, 3, 3, 1, 2, 3, 3, 1, 2, 2, 1, 3, 3, 2, 1),$

- при $k = 1$ мы получаем 3 различных заготовки: $(1), (2), (3),$
- при $k = 2$ — 6 различных заготовок: $(1, 1), (1, 2), (2, 2), (3, 3), (3, 1), (2, 3),$
- при $k = 3$ — 5 различных заготовок: $(1, 1, 1), (2, 2, 2), (3, 3, 3), (1, 2, 3), (3, 1, 2),$
- при $k = 4$ — 5 различных заготовок: $(1, 1, 1, 2), (2, 2, 3, 3), (3, 1, 2, 3), (3, 1, 2, 2), (1, 3, 3, 2),$
- При больших значениях k получатся не более 3 различных заготовок.

Input

В первой строке входного файла задано целое число n ($1 \leq n \leq 200\,000$), обозначающее количество шариков, нанизанных на нитку первоначально. Во второй строке заданы n целых чисел a_i ($1 \leq a_i \leq n$), разделённых пробелами, обозначающие цвета соответствующих шариков (считая с начала нитки).

Output

В первой строке выходного файла выведите два целых числа: наибольшее количество различных заготовок, которые могут быть получены при оптимальном выборе k и количество l таких k . Во второй строке в произвольном порядке выведите l попарно различных целых чисел — соответствующие значения k .

Examples

Standard input	Standard output
21	6 1
1 1 1 2 2 2 3 3 3 1 2 3 3 1 2 2 1 3 3	2
2 1	

Problem B. Bridges

Input file: Standard input
Output file: Standard output
Time limit: 2 seconds
Memory limit: 256 mebibytes

Прибрежный город Байтсберг состоит из n небольших островков, ныне плотно заселённых. Острова пронумерованы от 1 до n . Некоторые острова соединены мостами, по которым возможно двустороннее движение. Никакие два острова не соединены более, чем одним мостом, при этом с любого острова можно доехать до любого другого по системе мостов.

Посетивший Байтсберг математик Лео собирается объехать его на велосипеде. Его маршрут начнётся на острове с номером 1, пройдёт по каждому острову, по одному разу пройдёт по каждому мосту и закончится снова на острове с номером 1.

При этом подъём на каждый мост имеет определённый уровень сложности (при проезде в разных направлениях уровень сложности различен — например, мосты могут иметь различный наклон).

Помогите Лео составить соответствующий маршрут, при этом в случае, если маршрутов несколько, выберите тот, в котором максимальный уровень сложности подъёма на мост минимален.

Input

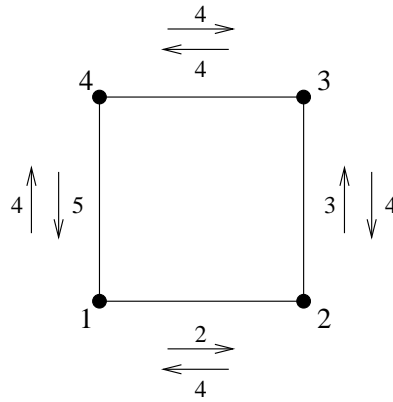
В первой строке входного файла заданы два целых числа n и m ($2 \leq n \leq 1000$, $1 \leq m \leq 2000$) — количество островов и количество мостов Байтсберга соответственно. Острова пронумерованы от 1 до n , мосты — от 1 до m . Последующие m строк задают мосты. $(i + 1)$ -я строка содержит четыре целых числа a_i, b_i, l_i, p_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq l_i, p_i \leq 1000$) — описание моста с номером i . Мост соединяет острова a_i и b_i . При этом «сложность» подъёма при проезде с a_i на b_i равна l_i , а при проезде с b_i на a_i — p_i .

Output

Если планируемое Лео путешествие невозможно, выведите в единственной строке выходного файла слово NIE (*нет* по-польски). В другом случае выходной файл должен состоять из двух строк, задающих требуемый маршрут. В первой строке выведите одно число — максимальный уровень сложности подъёма на мост для данного маршрута. Во второй строке выведите m целых чисел — номера мостов в получившемся маршруте, перечисленные в порядке их объезда. Если таких маршрутов несколько, выведите любой. *arbitrarily*.

Examples

Standard input	Standard output
4 4	4
1 2 2 4	4 3 2 1
2 3 3 4	
3 4 4 4	
4 1 5 4	



Explanation of the example:

Оптимальным маршрутом для Лео будет следующий:

$$1 \xrightarrow{4} 4 \xrightarrow{4} 3 \xrightarrow{4} 2 \xrightarrow{4} 1.$$

Максимальный уровень сложности подъёма на мост для этого маршрута равен 4.

Problem C. Frog

Input file: Standard input
Output file: Standard output
Time limit: 2.5 seconds
Memory limit: 256 mebibytes

На дне длинного прямого ручья лежат n камней, выступающих над поверхностью воды.

Их расстояния от истока ручья равны $p_1 < p_2 < \dots < p_n$ соответственно. Лягушка прыгает с камня на камень следующим образом. Каждый раз она прыгает на камень, который является k -м ближайшим к тому, на котором она сидит. Например, если лягушка сидит на камне с координатой p_i , p_j определяется следующим образом:

$$|\{p_a : |p_a - p_i| < |p_j - p_i|\}| \leq k \quad \text{and} \quad |\{p_a : |p_a - p_i| \leq |p_j - p_i|\}| > k.$$

Если p_j можно выбрать несколькими способами, выбирается камень, ближайший к истоку ручья.

По заданному исходному расположению лягушки вычислите, на каком камне она будет сидеть после m прыжков.

Input

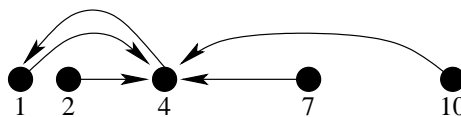
В первой строке входного файла содержатся три целых числа n , k и m ($1 \leq k < n \leq 1\,000\,000$, $1 \leq m \leq 10^{18}$), — количество камней, параметр k и количество предполагаемых прыжков. Вторая строка содержит n целых чисел p_j ($1 \leq p_1 < p_2 < \dots < p_n \leq 10^{18}$) — расстояния от истока ручья до соответствующих камней.

Output

В единственной строке выходного файла выведите n чисел. i -е число обозначает номер (начиная с 1) камня, на котором лягушка будет сидеть после m прыжков, начиная с i -го камня.

Examples

Standard input	Standard output
5 2 4	1 1 3 1 1
1 2 4 7 10	



На иллюстрации показано, куда лягушка прыгнет с каждого камня.

Problem D. Godzilla (Division 1 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 2 seconds
Memory limit: 256 mebibytes

Сеть кабельного телевидения в Бейтландии состоит из n узлов и m **односторонних** соединений. Сигнал передаётся на некоторые узлы (называемые *передающими* узлами). Остальные узлы могут принимать сигнал, если существует (не обязательно прямое) соединение, ведущее из передающего узла в данный узел.

Передающие узлы выбираются так, чтобы, во-первых, обеспечить целостность сети (то есть чтобы все узлы сети могли принимать сигнал), и, во-вторых, количество передающих узлов было бы минимально.

Однажды в стране появился злобный монстр Годзилла, который каждый день приводит в негодность по одному соединению. Поэтому для сохранения целостности сети каждый день набор передающих узлов выбирается заново. Требуется написать программу, которая вычисляла бы наименьшее возможное количество передающих узлов для каждого из дней после появления монстра.

Input

В первой строке входного файла заданы два целых числа n и m ($1 \leq n, m \leq 100\,000$) — соответственно количество узлов в кабельной сети и количество соединений. В следующих m строках описываются соединения: каждая из этих строк содержит два целых числа a и b ($1 \leq a, b \leq n, a \neq b$) и описывает однонаправленное соединение из узла a в узел b . Между любыми двумя узлами в одном направлении существует не более одного соединения. Следующая строка содержит целое число k ($1 \leq k \leq m$) — количество соединений, которые оказались повреждены. Далее следуют k строк. i -я из них содержит номер (в порядке перечисления во входном файле, начиная с 1) соединения, повреждённого в i -й день. При этом все повреждённые узлы попарно различны.

Output

Выведите k строк, i -я из которых содержит минимальное количество передающих узлов, необходимых для обеспечения целостности сети на i -й день после появления Годзиллы.

Examples

Standard input	Standard output
3 2	2
1 2	3
2 3	
2	
2	
1	

Problem E. Intelligence test

Input file: Standard input
Output file: Standard output
Time limit: 2.5 seconds
Memory limit: 256 mebibytes

Одна из частей теста IQ, принятого в Бейтландии, состоит в следующем: необходимо вычеркнуть из заданной последовательности несколько чисел определённым образом.

Ваша задача — написать программу автоматической проверки того, что некоторая последовательность может быть получена из заданной путём вычёркивания некоторого (возможно, нулевого) количества чисел.

Input

Первая строка входного файла содержит целое число m ($1 \leq m \leq 1\,000\,000$) — длину первоначальной последовательности. В следующей строке заданы m целых чисел a_1, a_2, \dots, a_m ($1 \leq a_i \leq 1\,000\,000$ для $1 \leq i \leq m$) — элементы первоначальной последовательности. В третьей строке записано целое число n — количество последовательностей, вложение которых нужно проверить. Последующие $2n$ строк содержат описания этих последовательностей. Каждая последовательность описывается двумя строками. Первая из них содержит целое число m_i ($1 \leq m_i \leq 1\,000\,000$) — длину последовательности. Вторая содержит последовательность из m_i элементов: $b_{i,1}, b_{i,2}, \dots, b_{i,m_i}$ ($1 \leq b_{i,j} \leq 1\,000\,000$ для $1 \leq j \leq m_i$). Суммарная длина всех n последовательностей не превосходит $1\,000\,000$.

Output

Для каждой из n последовательностей выведите в отдельной строке в порядке следования во входном файле слово “ТАК” (*да* по-польски) если соответствующая последовательность может быть получена вычёркиванием некоторого количества чисел из первоначальной, или “NIE” (*нет* по-польски) в противном случае.

Examples

Standard input	Standard output
7	TAK
1 5 4 5 7 8 6	NIE
4	TAK
5	NIE
1 5 5 8 6	
3	
2 2 2	
3	
5 7 8	
4	
1 5 7 4	

Problem F. Lamp (Division 1 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 6 seconds
Memory limit: 256 mebibytes

Здания B и C стоят друг напротив друга на расстоянии 10 метров параллельно друг другу. При этом n окон здания B «смотрят» на здание C и m окон здания C «смотрят» на здание B .

Ночью у основания первого здания строители включили мощный прожектор и один из жителей здания B , известный адвокат, проснулся от того, что отражённый свет прожектора попал в его окно. Разозлившись, адвокат сел писать жалобу на строителей от лица всех пострадавших жильцов здания B . Для составления текста адвокату требуется знать, какие ещё окна здания B оказались освещены аналогичным образом. Напишите программу, это вычисляющую.

Для решения задачи примем, что прожектор — это точка, а его свет распространяется строго согласно законам геометрической оптики. Напомним, что свет распространяется по прямой, и если *луч попадает в окно*, он отражается, при этом угол падения равен углу отражения.

Введём координатную систему на обращённых друг к другу стенах двух зданий следующим образом. Обе оси X горизонтальны, обе оси Y — вертикальны; оси на обеих стенах ориентированы одинаково и точки $(0, 0)$ лежат одна напротив другой.

Окна в каждом из зданий — прямоугольники со сторонами, параллельными осям системы координат. При этом никакие два окна не имеют общей внутренней точки. Если луч попадает на границу окна, то он не отражается. Прожектор находится на стене здания B в точке $(0, 0)$, которая не принадлежит никакому окну или его границе.

Input

В первой строке входного файла заданы два целых числа n и m ($1 \leq n, m \leq 600$) — количество окон в зданиях B и C соответственно. Последующие n строк описывают окна здания B . Строка с номером $i + 1$ (для $1 \leq i \leq n$) содержит четыре целых числа $x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i}$ ($-1000 \leq x_{1,i} < x_{2,i} \leq 1000$, $0 \leq y_{1,i} < y_{2,i} \leq 1000$) — координаты левого нижнего и правого верхнего углов i -го окна.

В последующих m строках аналогичным образом заданы окна здания C . Все расстояния заданы в метрах.

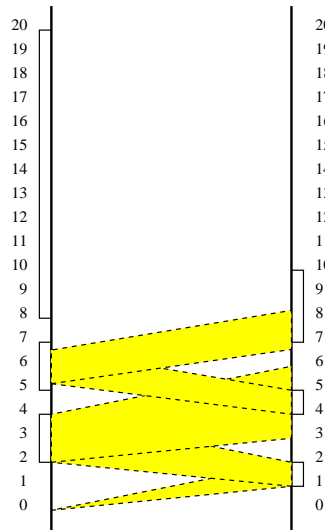
Output

В первой строке выходного файла выведите количество окон здания B , во внутренней точке которых попадает отражение луча от прожектора. Гарантируется, что как минимум одно такое окно существует (окно адвоката).

Во второй строке выведите номера этих окон (окна пронумерованы в порядке, заданном во входном файле, начиная с 1), отсортированные по возрастанию.

Examples

Standard input	Standard output
3 3 -1 2 1 4 -1 5 1 7 -3 8 -2 20 -1 1 1 2 -1 4 1 5 -1 7 1 10	2 1 2



Example explanation: Луч света попадает в первое окно здания B после отражения от первого окна здания C (например, после отражения в точке $(0, 1.5)$ здания C он попадает в точку $(0, 3)$ здания B). Для того, чтобы попасть во второе окно здания B , луч отражается трижды — тот же самый луч попадает в точку $(0, 4.5)$ внутри второго окна здания C , и затем в точку $(0, 6)$ внутри второго окна здания B . В третье окно здания B свет не попадает. Хотя каждая точка стены здания C освещена каким-то лучом, на рисунке обозначены только те лучи, которые после отражения освещают какие-то окна здания B .

Problem G. Leonardo's Numbers (Division 1 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 6 seconds
Memory limit: 256 mebibytes

Определим последовательность (L_i) чисел Леонардо следующим образом:

$$L_0 = L_1 = 1,$$
$$L_{i+1} = L_i + L_{i-1} + 1 \quad \text{for } i \geq 1.$$

Требуется найти значение следующего выражения

$$L_0^k + L_1^k + L_2^k + \dots + L_n^k$$

для заданных n и k .

Input

Входной файл содержит два целых положительных числа n и k ($k \leq 13$, $0 \leq n \leq 2^{64} - 1$).

Output

В выходной файл выведите 9 последних цифр десятичного представления требуемой суммы.

Examples

Standard input	Standard output
3 2	000000036

Требуемая сумма равна $1^2 + 1^2 + 3^2 + 5^2 = 36$.

Problem H. Monotonicity (Division 1 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 3 seconds
Memory limit: 256 mebibytes

Для последовательности целых чисел a_1, a_2, \dots, a_n определим её *схему монотонности* как последовательность s_1, s_2, \dots, s_{n-1} , состоящую из символов $<$, $>$ и $=$. Символ s_i строится как знак отношения между a_i и a_{i+1} . Например, для последовательности 2, 4, 3, 3, 5, 3 схема монотонности выглядит как $<, >, =, <, >$.

Считается, что последовательность b_1, b_2, \dots, b_{n+1} со схемой монотонности s_1, s_2, \dots, s_n *реализует* другую схему монотонности s'_1, s'_2, \dots, s'_k , если для каждого $i = 1, 2, \dots, n$ она содержит $s_i = s'_{((i-1) \bmod k)+1}$.

Иначе говоря, последовательность s_1, s_2, \dots, s_n может быть получена путём повторения последовательности s'_1, s'_2, \dots, s'_k и удаления соответствующего суффикса из результата. Например, последовательность 2, 4, 3, 3, 5, 3 реализует, наряду со многими другими, следующие схемы:

- $<, >, =$
- $<, >, =, <, >$
- $<, >, =, <, >, <, <, =$
- $<, >, =, <, >, =, >, >$

Вам заданы целочисленная последовательность a_1, a_2, \dots, a_n и схема монотонности s_1, s_2, \dots, s_k . Ваша задача — найти наибольшую подпоследовательность $a_{i_1}, a_{i_2}, \dots, a_{i_m}$ ($1 \leq i_1 < i_2 < \dots < i_m \leq n$) заданной последовательности, которая реализует заданную схему.

Input

В первой строке входного файла заданы два целых числа n и k ($1 \leq n \leq 500\,000$, $1 \leq k \leq 500\,000$) — длины последовательности (a_i) и схемы монотонности (s_j) соответственно.

Во второй строке входного файла задана последовательность (a_i) , то есть n целых чисел a_i ($1 \leq a_i \leq 1\,000\,000$).

Третья строка содержит схему монотонности (s_j) , то есть k символов s_j (каждый символ может быть или $<$, или $>$, или $=$).

Output

В первой строке выходного файла выведите целое число m — наибольшая длина подпоследовательности a_1, a_2, \dots, a_n , реализующей заданную схему s_1, s_2, \dots, s_k .

Во второй строке выведите m чисел $a_{i_1}, a_{i_2}, \dots, a_{i_m}$ — соответствующую подпоследовательность. Если таких подпоследовательностей несколько, выведите любую из них.

Examples

Standard input	Standard output
7 3	6
2 4 3 1 3 5 3	2 4 3 3 5 3
< > =	

Problem I. Sheep (Division 1 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 3.5 seconds
Memory limit: 256 mebibytes

Пастбища в Байтландии представляют собой выпуклые многоугольники. При этом у каждой из овец на пастбище есть «своя точка» строго внутри (не на сторонах) пастбища — место, где она проводит практически всё время. Также иногда овцы устраивают поединки друг с другом. Байтландские пастухи считают, что это полезно для укрепления навыков самозащиты от хищников, поэтому в стаде каждого пастуха количество овец чётно, чтобы каждая овца могла найти себе соперника для поединка.

В одном байтландском селе в соответствии с рекомендациями специалистов решили провести следующий эксперимент: пастбище разделили на треугольники с помощью $n - 3$ внутренних перегородок, соединяющих вершины многоугольника и не пересекающихся нигде, кроме этих вершин. При этом ни одна из перегородок не должна проходить в точности через место, в котором обычно находится какая-то овца, и в каждом из получившихся треугольников должно быть чётное количество «своих точек», чтобы сохранить возможность поединков.

Требуется вычислить количество подобных разбиений первоначального пастбища.

Input

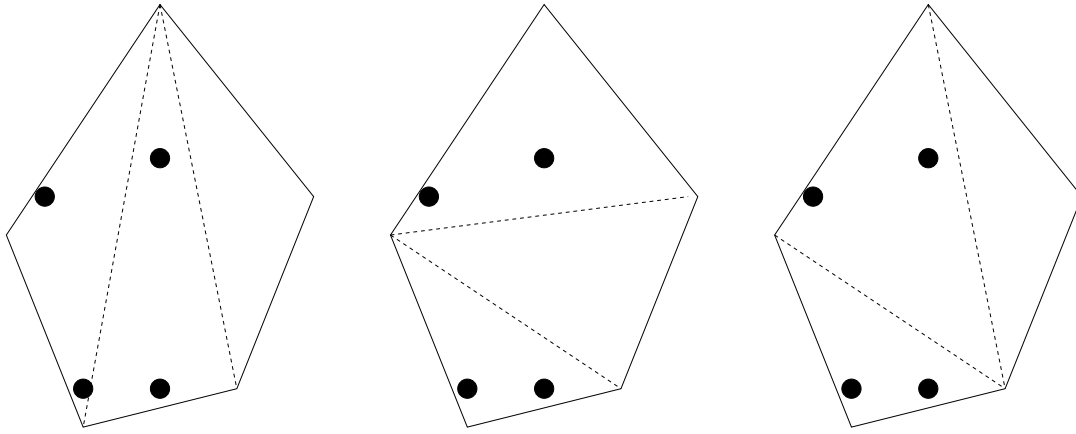
Первая строка входного файла содержит три целых числа n , k и m ($4 \leq n \leq 600$, $2 \leq k \leq 20\,000$, $2 \mid k$, $2 \leq m \leq 20\,000$) — количество вершин многоугольника, образующего пастбище, количество овец и модуль m , который потребуется для вывода ответа. Каждая из последующих n строк содержит два целых числа x_i и y_i ($-15\,000 \leq x_i, y_i \leq 15\,000$) — координаты i -й вершины пастбища. Вершины заданы в порядке обхода по часовой стрелке. Каждая из последующих k строк содержит два целых числа p_j , q_j ($-15\,000 \leq p_j, q_j \leq 15\,000$) — координаты «своей точки» для j -й овцы.

Output

Выведите одно целое число — остаток от деления количества описанных в условии разбиений на m .

Examples

Standard input	Standard output
5 4 10 5 5 3 0 -1 -1 -3 4 1 10 1 0 -1 0 1 6 -2 5	3



На иллюстрации приведены все три разбиения пастбища на треугольники. Также на карте обозначены «свои точки» всех овец.

Problem J. Teleportation

Input file: Standard input
Output file: Standard output
Time limit: 2 seconds
Memory limit: 256 mebibytes

Обитаемая планетная система около Тау Кита устроена следующим образом: планеты занумерованы числами от 1 до n , на планете с номером 1 находится резиденция Всеобщего Правительства, на планете с номером 2 — космодром, с которого стартуют межгалактические корабли. Между планетами 1 и 2 установлен двусторонний правительственный портал, перемещающий с одной планеты на другую за 250 минут.

Однако развитие технологий телепортации привело к тому, что стала возможна постройка общественных порталов, перемещающих с некоторой планеты на другую за 1 час. При этом между двумя планетами можно построить не более одного общественного портала. Некоторые планеты уже соединены общественными порталами, но при этом в целях антитеррористической защиты система устроена так, что с планеты 1 до планеты 2 нельзя добраться быстрее, чем через правительственный портал. Время пересадки при этом считается нулевым.

Какое максимальное количество общественных порталов можно построить дополнительно к имеющимся так, чтобы требования антитеррористической защиты не были нарушены?

Input

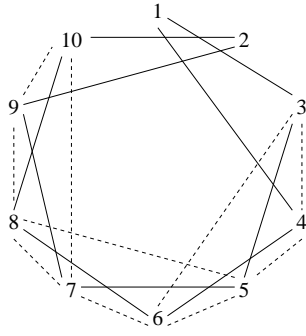
Во входном файле заданы два целых числа n и m ($2 \leq n \leq 40\,000$, $0 \leq m \leq 1\,000\,000$) — количество планет в системе и количество уже построенных общественных порталов. Следующие m строк описывают общественные порталы следующим образом: каждая из них содержит два целых числа a_i и b_i ($1 \leq a_i < b_i \leq n$) — номера планет, соединённых порталами. При этом никакие две планеты не соединены порталом дважды. Гарантируется, что между планетами 1 и 2 можно добраться по сети существующих общественных порталов, при этом время в пути составит не менее 250 минут.

Output

Выведите одно целое число — наибольшее количество общественных порталов, достраиваемых в дополнение к существующим без нарушения требований антитеррористической защиты.

Examples

Standard input	Standard output
10 10 1 3 3 5 5 7 7 9 2 9 1 4 4 6 6 8 8 10 2 10	10



Существующие порталы обозначены непрерывными линиями, достраиваемые без нарушения требований антитеррористической защиты — пунктирными.

Problem K. Ancestry (Division 2 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 2 seconds
Memory limit: 256 Mebibytes

При определении порядка наследования в Бйтландии важную роль играет определение «линии наследования».

Если A_0 является дальним потомком B_0 , жившего столетиями ранее, то линия наследования не обязательно одна — например, дальние родственники могли вступить в брак, уже не зная о родственных связях. В этом случае часто бывает важно найти линию наследования от B_0 к A_0 , содержащую минимальное количество женщин (наследование в Бйтландии чаще всего происходит по мужской линии).

Вам задана база данных с информацией по некоторым людям и их предкам. Ваша задача — найти линию наследования от B_0 к A_0 , содержащую наименьшее количество женщин.

Input

Входной файл имеет следующий формат:

- В первой строке задано одно целое число N ($2 \leq N \leq 10^5$) — количество человек, информация о которых есть в базе данных.
- Во второй строке заданы два различных числа A_0 и B_0 ($1 \leq A_0, B_0 \leq N$) — идентификаторы в базе данных для двух человек, информация по которым нас интересует.
- В следующих N строках заданы по два целых числа f и m ($0 \leq f, m \leq N$); в i -той строке заданы идентификаторы отца (f) и матери (m) для человека с идентификатором, равным i . Если f или m равны 0, то информации про отца (или про мать, соответственно) этого человека в базе отсутствует.

Гарантируется, что база корректна: если кто-то записан хотя бы в одном случае в качестве отца, он не может быть записан в качестве матери, и наоборот; также информация о предках не содержит циклов.

Output

В выходной файл выведите одно число — наименьшее количество женщин в линии наследования от B_0 к A_0 . Если таковой линии наследования по базе установить не удалось, выведите `'no ancestor'`.

Example

Standard input	Standard output
23 1 8 2 0 3 15 9 4 5 10 11 6 0 7 8 0 0 0 0 0 0 0 12 14 13 0 0 0 0 0 16 23 17 20 0 18 19 0 12 7 21 0 22 0 8 0 0 0	2
9 2 9 5 2 3 4 0 0 0 0 6 7 0 0 8 9 0 0 0 0	no ancestor

Problem L. Bus (Division 2 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 2 seconds
Memory limit: 256 Mebibytes

Требуется написать программу для решения задач, подобных следующей:

Автобус выезжает с начальной станции с 4 пассажирами. На первой остановке садятся 3 пассажира, на следующей 1 пассажир выходит и двое садятся. Сколько пассажиров находятся в автобусе на данный момент?

Входной файл будет описывать сценарий подобной задачи. От вашей программы требуется вычислить, сколько пассажиров будет в автобусе в конце сценария.

Input

Входной файл начинается с номера маршрута (от 1 до 5 латинских букв или цифр) и числа Z — количества посадочных мест в автобусе ($10 \leq Z \leq 100$), разделённых пробелом. Во второй строке задано целое число P — первоначальное количество пассажиров в автобусе ($0 \leq P \leq Z$). Третья строка содержит S , количество остановок на маршруте ($1 \leq S \leq 100$). В каждой из последующих S строк содержатся 2 целых числа, разделённых пробелами. Первое число задаёт количество пассажиров, выходящих из автобуса, второе — количество пассажиров, ожидающих посадки. Если количество пассажиров, желающих уехать, больше количества оставшихся посадочных мест, то уезжают только те пассажиры, которым достались посадочные места (то есть стоя никто не едет).

Output

В выходной файл выведите номер маршрута в том виде, в котором он был задан в выходном файле, и количество пассажиров в автобусе в конце заданного сценария.

Example

Standard input	Standard output
28HZC 26 4 2 0 3 1 2	28HZC 8

Problem M. Circles (Division 2 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 2 seconds
Memory limit: 256 Mebibytes

На плоскости задано несколько кругов с целочисленными радиусами и координатами центров. Все точки плоскости, обе координаты которых являются целыми числами и находятся между $-(2^{14} - 1)$ и 2^{14} включительно, окрашены в красный цвет.

Вычислите количество красных точек, лежащих внутри или на границе объединения заданных кругов.

(В первом тестовом примере круг с радиусом 1 и центром (2, 2) содержит 5 красных точек, круг с радиусом 2 и центром (3, 3) содержит 13 красных точек, их объединение содержит 15 красных точек).

Input

Входной файл состоит из $1 \leq n \leq 10^4$ строк, где n — количество кругов. Каждая строка содержит три целых числа: координаты центра окружности x и y и радиус r ($-2 \cdot 10^4 \leq x, y \leq 2 \cdot 10^4$, $1 \leq r \leq 1000$).

Output

В выходной файл выведите одно целое число — количество красных точек, лежащих внутри или на границе объединения заданных кругов.

Example

Standard input	Standard output
2 2 1 3 3 2	15
-16383 -16383 2	6

Problem N. Stars (Division 2 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 2 seconds
Memory limit: 256 Mebibytes

Найти определённое созвездие на небе бывает не так просто даже с атласом созвездий. Более того, конфигурации созвездий, состоящие из небольшого количества звёзд, могут быть найдены несколько раз, и отличить правильное бывает не так просто.

Вам задаётся карта звёздного неба (представленная в виде набора точек на плоскости, для каждой из которых определена «яркость»). Также вам заданы созвездия, также в виде набора точек на плоскости. Ваша задача — определить, сколько раз созвездие встретится на звёздной карте, и, если среди встретившихся «кандидатов» есть одно с наибольшей яркостью — вывести его координаты.

Считается, что созвездие встретилось на карте, если существует подмножество звёзд, которое переводится в созвездие комбинацией поворота и масштабирования (возможно, нулевых). При этом в случае, если разные преобразования дают совпадающие подмножества (например, если созвездие имеет форму квадрата, то это повороты на 0, 90, 180 и 270 градусов), считается, что созвездие всё равно встретилось один раз.

Яркость встретившегося созвездия определяется как средняя яркость всех звёзд, его составляющих.

Input

В первой строке входного файла задано целое число N — количество звёзд на карте ($1 \leq N \leq 1000$). Каждая из последующих N строк содержит три целых числа — X - и Y -координаты и яркость для каждой звезды. Чем больше значение, тем ярче звезда.

Следующая строка содержит целое число M — количество созвездий, которые надо найти ($1 \leq M < 50$).

Описание каждого созвездия начинается со строки, содержащей целое число S — количество звёзд в созвездии и строки C — названия созвездия (C состоит из не более, чем 40 символов с кодом, не меньшим 33 и не большим 127). Последующие S строк содержат по два целых числа — X - и Y -координаты звёзд созвездия.

Все координаты не превосходят 1000 по абсолютной величине, яркость может принимать значения от 0 до 100.

Output

Для каждого созвездия в порядке, указанном во входном файле, выведите название созвездия, а также число, показывающее, сколько раз созвездие встретилось на карте (в формате, определяемом примером). Если созвездие встретилось на карте как минимум 1 раз, выведите позицию наиболее яркого «кандидата» — список координат звёзд, составляющих это вхождение. Звёзды перечисляются по возрастанию координаты X , а при равенстве X — по возрастанию Y . Гарантируется, что такое вхождение единственно. Формат вывода определяется примером. Между описаниями двух созвездий выводите пустую строку.

Example

Standard input	Standard output
6 1 2 1 2 1 4 2 4 3 3 2 1 4 1 5 4 3 2 2 3 Triangulum 1 1 3 1 2 4 4 Cancer 1 3 4 3 6 1 7 5	Triangulum occurs 2 time(s) in the map. Brightest occurrence: (1,2) (4,1) (4,3) Cancer occurs 0 time(s) in the map.

Problem O. IOI team (Division 2 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 2 seconds
Memory limit: 256 Mebibytes

Министерство образования Байтландии объявило новый порядок выбора команды на IOI. Тренер объявляет число k , после чего все победители локальной олимпиады по информатике становятся в круг и получают в порядке обхода номера $1, 2, \dots, n$ (при этом, в частности, участник с номером n соседствует с участником с номером 1). После чего тренер отсчитывает по кругу k участников, начиная с первого, удаляет k -го, далее снова отсчитывает k участников, удаляет k -го и так далее, пока в круге остаётся хотя бы один человек. Последние 4 удалённых из круга участника и поедут на IOI.

По заданным n и k вычислите номера участников, которые поедут на IOI от Байтландии.

Input

Во входном файле содержатся два целых числа n — общее количество победителей локальной олимпиады, участвующих в отборе и k — количество участников, которое отсчитывает тренер ($4 \leq n \leq 10^8$, $1 \leq k \leq 10^8$). student.

Output

В выходной файл выведите 4 числа — номера участников, которые поедут на IOI от Байтландии, перечисленные в порядке, в котором тренер удалял их из круга.

Example

Standard input	Standard output
9 3	5 2 7 1
11 7	6 1 4 5