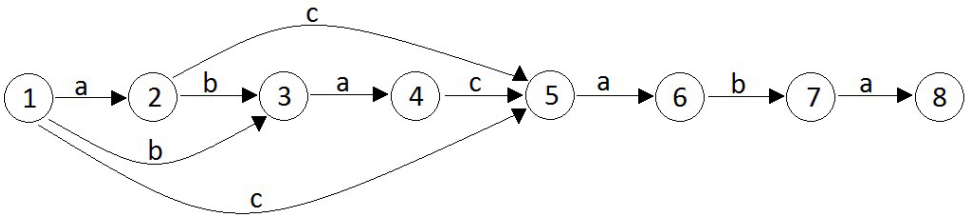
Problem A. Automaton

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 секунды
Memory limit:	256 мебибайт

Дана строка S. Построить детерминированный конечный автомат, принимающий все суффиксы строки S (и возможно другие конечные строки). Автомат должен состоять из минимального числа состояний — N и не более чем $2 \cdot N$ переходов. Каждое состояние автомата объявляется финальным. Начальное состояние имеет номер 1.

На изображении ниже показан автомат из тестового примера для строки "`abacaba`"



Input

В единственной строке слово S, состоящее из строчных латинских букв.

Output

В первой строке два числа N и K — количество состояний и количество переходов. Далее K строк, в каждой из которых по два числа a_i, b_i и буква c_i, означающие наличие перехода из состояния a_i в b_i по букве c_i. Переходы можно выводить в любом порядке. Если решений несколько, можете вывести любое из них.

Limits

$1 \leqslant |S| \leqslant 10^5$

$1 \leqslant K \leqslant 2N$

$1 \leqslant a_i, b_i \leqslant N$

'a' $\leqslant c_i \leqslant$ 'z'

Example

standard input	standard output
abacaba	8 10 1 2 a 1 3 b 1 5 c 2 3 b 2 5 c 3 4 a 4 5 c 5 6 a 6 7 b 7 8 a

Problem B. Beinz (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 секунда
Memory limit:	256 мебибайт

k-мерный симплекс Паскаля — это такая k-мерная фигура, что в ячейке $(i_1, i_2, ..., i_k)$ записано число $C_n^{i_1,i_2,...,i_k}$, где $i_j \geq 0$, а $n = i_1 + i_2 + ... + i_k$.

Пример:

1-мерный симплекс Паскаля — это бесконечная строка из единиц

2-мерный симплекс Паскаля — это обычный треугольник Паскаля

3-мерный симплекс Паскаля — это бесконечный тетраэдр, в котором записаны сочетания вида $C_n^{i,j,k}$ и так далее.

Допустим, что у нас есть k-мерный симплекс Паскаля, в котором все сочетания записаны по модулю p (p — простое). Рассмотрим n-тый слой этого симплекса (тот, где записаны все сочетания из n). Необходимо найти количество ненулевых элементов на этом слое. Так как это число может оказаться большим, необходимо вывести его по модулю $10^9 + 7$.

Input

В первой строке даны три числа k, p и t. В каждой из следующих t строк находится одно число n_i — номер слоя.

Output

Ровно t строк, в каждой из которых для соответствующего n_i вывести количество ненулевых элементов на n_i-том слое симплекса, по модулю $10^9 + 7$.

Limits

$1 \leqslant k \leqslant 10^3$

$2 \leqslant p \leqslant 10^6 + 3$, p — простое.

$1 \leqslant t \leqslant 10^5$

$0 \leqslant n_i \leqslant 10^{18}$

Example

standard input	standard output
2 7 4 0 6 7 8	1 7 2 4
3 7 4 0 6 7 8	1 28 3 9

Problem C. Cutting (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 секунды (*3 секунды для Java*)
Memory limit:	256 мебибайт

Дана строка S и список M, состоящий из N слов. За одну операцию можно выбрать некоторую подстроку строки S, если такая встречается в качестве слова в списке M, и вырезать из строки S. После чего оставшиеся части строки S, если такие есть, склеиваются. Определить, за какое минимальное количество операций можно уничтожить всю строку S. Гарантируется, что это можно сделать.

Input

В первой строке слово S. Во второй строке целое число N — количество слов в списке. Далее N строк, в каждой из которых по одному слову из списка M. Все слова состоят только из строчных латинских букв.

Output

Одно число — минимальное количество операций, необходимое для уничтожения строки S.

Limits

$1 \leqslant |S| \leqslant 100$

$1 \leqslant N \leqslant 100$

$1 \leqslant |M_i| \leqslant 100$

Example

standard input	standard output
abacaba 4 aba aca a b	3

Explanation

Первой операцией вырезали подстроку “aca”, получили “abba”.

Второй операцией вырезали подстроку “b”, получили “aba”.

Третьей операцией удалили всю строку “aba”.

Problem D. Disclosure (Division 1 Only!)

Input file:	standard input
Output file:	standard output
Time limit:	1 секунда
Memory limit:	512 мебибайт

You are given an acyclic directed graph, consisting of N nodes and K edges. Delete maximum amount of edges, without causing graph's transitive closure to change.

Transitive closure of graph G is a graph G', consisting of all nodes of orgignal graph G and such edges (u, v), that there is a path from node u to v in graph G.

Input

The first line contains two numbers N and K. Then K lines follow, each containing two integers a_i and b_i, descripting directed edge from a_i to b_i. Graph doesn't contain loops, cycles and multiple edges.

Output

Output the every edge that is necessary to remain in graph as the pair of nodes connected by this edge. Edges can be written in any order.

Limits

$1 \leqslant N \leqslant 50000$

$0 \leqslant K \leqslant 50000$

$1 \leqslant a_i, b_i \leqslant N$

Example

standard input	standard output
5 6 1 2 2 3 3 5 4 5 1 5 1 3	1 2 2 3 3 5 4 5

Problem E. Embedded circles (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 секунды
Memory limit:	256 мебибайт

Дана таблица $\{a_{ij}\}$ из R строк и C столбцов, состоящая из цифр от '0' до '9'. Требуется ответить на Q запросов вида:

i_k j_k r_k, $1 \leqslant k \leqslant Q$

Найти сумму всех элементов таблицы a_{ij} таких, что $(i - i_k)^2 + (j - j_k)^2 \leqslant r_k^2$.

Область суммирования для каждого запроса представляет собой круг с центром в ячейке (i_k, j_k) и радиусом r_k. Область определяется в точности по формуле выше, но для удобства будем называть ее кругом.

Для любых двух запросов выполняется следующее: либо их круги не имеют общих ячеек таблицы, либо один из кругов полностью содержится в другом.

Более того, запросы идут в порядке вложенности. Это означает, что если круги запросов k и l $(k < l)$ имеют общие ячейки, то из этого следует, что для любого t: $k < t \leqslant l$, круг t полностью содержится в круге запроса k. Круги различных запросов могут совпадать.

Input

В первой строке два числа R и C — размеры таблицы. Далее R строк по C цифр в каждой (между цифрами нет пробелов). В следующей строке число Q — количество запросов. Далее в Q строках описаны запросы по три целых числа: i_k, j_k, r_k, в каждой строке — номера строки и столбца центральной ячейки и радиус круга. Ячейки нумеруются с 1.

Output

Поскольку запросов много, выведите одно число — сумму всех ответов на запросы.

Limits

$1 \leqslant R, C \leqslant 2\,000$

'0' $\leqslant a_{ij} \leqslant$ '9'

$1 \leqslant Q \leqslant 10^6$

$1 + r_k \leqslant i_k \leqslant R - r_k$

$1 + r_k \leqslant j_k \leqslant C - r_k$

$0 \leqslant r_k \leqslant \dfrac{\min(R, C) - 1}{2}$

Example

standard input	standard output
6 6 123456 234567 345678 456789 567890 678901 10 1 1 0 3 3 2 3 2 1 2 2 0 4 2 0 1 3 0 2 3 0 4 3 0 5 5 1 5 5 0	141

Explanation

$141 = 1 + 65 + 20 + 3 + 5 + 3 + 4 + 6 + 25 + 9$

Problem F. False figures (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 секунды (*3 seconds for Java*)
Memory limit:	256 мебибайт

Эта задача является валидатором тестов к предыдущей задаче. Самое сложное уже проверили, вам предлагается всего лишь проверить вложенность кругов.

Представим таблицу $\{a_{ij}\}$ из 1 000 строк и 1 000 столбцов. Дано Q запросов вида:

i_k j_k r_k, $1 \leqslant k \leqslant Q$

Каждый из них определяет область из элементов таблицы a_{ij} таких, что $(i - i_k)^2 + (j - j_k)^2 \leqslant r_k^2$.

Для удобства будем называть эту область кругом с центром в ячейке (i_k, j_k) и радиусом r_k.

Пара запросов k и l $(k < l)$ считается невалидной, если круги k и l имеют общие ячейки, и существует круг t: $k < t \leqslant l$, который не содержится в круге k. Круги различных запросов могут совпадать.

Определить, есть ли среди запросов хотя бы одна невалидная пара.

Input

В первой строке число Q — количество запросов. Далее в Q строках описаны запросы по три целых числа: i_k, j_k, r_k, в каждой строке — номера строки и столбца центральной ячейки и радиус круга. Ячейки нумеруются с 1.

Output

Если есть хотя бы одна невалидная пара запросов, вывести номера этих запросов в произвольном порядке. Если таких пар много, разрешается вывести любую из них. Если невалидных пар нет, вывести “Ok”.

Limits

$1 \leqslant Q \leqslant 10^6$

$1 + r_k \leqslant i_k \leqslant 1\,000 - r_k$

$1 + r_k \leqslant j_k \leqslant 1\,000 - r_k$

$0 \leqslant r_k < 500$

Example

standard input	standard output
6 10 10 5 10 10 4 5 10 0 10 5 0 10 15 0 15 10 0	Ok
2 6 6 5 11 11 5	2 1

Problem G. Grouping (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 секунды
Memory limit:	256 мебибайт

Дано N целых чисел $\{a_i\}$. Пусть $\{b_j\}$ — такие K чисел (не обязательно целых), что:

$$S = \sum_{i=1}^{N} \min_{1 \leqslant j \leqslant K} |a_i - b_j|$$

И S — минимально возможное.

Найти S.

Input

В первой строке два числа N и K. Во второй строке ровно N целых чисел — $\{a_i\}$.

Output

Единственное вещественное число — S с абсолютной или относительной погрешностью не более 10^{-8}.

Limits

$1 \leqslant N \leqslant 5\,000$

$1 \leqslant K \leqslant N$

$0 \leqslant a_i \leqslant 400\,000$

Example

standard input	standard output
5 3 1 5 7 10 14	5.0

Explanation

В качестве $\{b_i\}$ имеет смысл взять $\{1, 7, 14\}$.

Problem H. Hidden triangles (Division 1 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 секунды
Memory limit:	512 мебибайт

На плоскость положили N треугольников по порядку от 1 до N. Вся внутренняя область каждого треугольника является непрозрачной и закрывает все, что находится под ней.

Определить, какие из треугольников остались видны на плоскости. То есть имеют область положительной площади, не накрытую сверху никаким другим треугольником.

Input

В первой строке число N — количество треугольников. Далее в N строках перечислены треугольники в том порядке, в котором они выкладывались на плоскость. Каждый треугольник описывается шестью целыми числами $x_{i1}, y_{i1}, x_{i2}, y_{i2}, x_{i3}, y_{i3}$ — координатами его вершин. Все треугольники невырожденные. Любая сторона одного треугольника имеет не более одной общей точки с любой стороной другого треугольника.

Output

В первой строке вывести количество видимых треугольников. Во второй строке перечислить их номера в произвольном порядке.

Limits

$1 \leqslant N \leqslant 500$

$-1\,000 \leqslant x_{ij}, y_{ij} \leqslant 1\,000$, для $1 \leqslant i \leqslant N, 1 \leqslant j \leqslant 3$

Example

standard input	standard output
3 1 0 4 0 0 3 -2 1 5 -2 3 4 -2 2 4 1 2 4	2 2 3

Problem I. Interactive

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 секунда
Memory limit:	256 мебибайт

http://codeforces.ru/blog/entry/4037

Alex_KPR: — ...начиная где-то с декартовых деревьев, FFT и так далее, вбивание и отладка кода становятся напрягающими и довольно занудными задачами...

Kunyavsky: — А можно я придерусь к мелочи. Что делают в одном ряду по времени написания декартово дерево и FFT?

Alex_KPR: — это первое, что пришло в голову, где нужно много писать и есть где накосячить

Kunyavsky: — Я же сказал: придираюсь к мелочам. Просто на мой взгляд рядом поставлены вещи, одна из которых несравнимо легче по написанию

homo_sapiens: — Полностью согласен Фурье пишется вдвое может даже втрое быстрее чем дерево (но его обычно пихать надо руками и ногами)

Kunyavsky: — Я имел ввиду с точностью наоборот. Все таки очень специфичная тема.

homo_sapiens: — Видимо и правда на вкус и цвет...

Пришло время узнать, что же проще.

Вам предлагаются две последовательности из N целых чисел $\{x_i\}$ и $\{y_i\}$.

Если вам больше нравится FFT, представьте, что это коэффициенты двух многочленов:

$$P(z) = \sum_{i=0}^{N-1} x_i z^i$$

$$Q(z) = \sum_{i=0}^{N-1} y_i z^i$$

Найдите коэффициенты многочлена $P(z) \cdot Q(z)$.

Если же вам больше нравятся Декартовы деревья, представьте, что $\{x_i\}$ — ключи, а $\{y_i\}$ — приоритеты. Постройте Декартово дерево путем вставки в него последовательно всех N элементов (x_i, y_i). После каждой вставки выводите высоту получившегося дерева. Высота дерева — количество ребер на длиннейшем пути от какой-либо вершины к корню.

Для тех, кто выбрал FFT, напомним, что Декартово дерево — это двоичное дерево поиска по ключам, содержащимся в вершинах (x_i). И одновременно куча по приоритетам в вершинах (y_i). То есть, для каждой вершины дерева выполняется свойство, что все вершины из левого поддерева имеют ключи меньшие, чем в данной вершине, а все вершины из правого поддерева — большие. А также приоритет данной вершины больше приоритета ее сыновей.

Независимо от того, что же вы выбрали, автор убедительно просит не использовать в этой задаче prewritten code.

Input

В первой строке число N. Во второй строке N различных целых чисел — $\{x_i\}$. В третьей строке N различных целых чисел — последовательность $\{y_i\}$.

Output

Если вы выбрали FFT, то выведите $2N-1$ целых чисел — коэффициенты результирующего многочлена. Если же вы выбрали Декартово дерево, то выведите N целых чисел — высоты дерева после вставки в него каждой вершины.

Вообще, вы можете вывести и то, и то, но тогда оба результата будут проверяться на корректность.

Limits

$1 \leqslant N \leqslant 50\,000$

$1 \leqslant x_i \leqslant 50\,000$

$1 \leqslant y_i \leqslant 50\,000$

гарантируется, что высота дерева не превысит 50

Example

standard input	standard output
6 4 1 2 3 6 5 1 6 3 4 5 2	4 25 20 34 54 71 72 58 56 37 10
6 4 1 2 3 6 5 1 6 3 4 5 2	0 1 2 2 3 4

Problem J. Journey

Input file:	standard input
Output file:	standard output
Time limit:	1 секунда
Memory limit:	256 мебибайт

Говорят, планета J-Pax имеет форму прямоугольного параллелепипеда с длиной A, шириной B и высотой C. Человеку, живущему в самой вершине, захотелось путешествовать. Он выбрал в качестве цели самую удаленную по поверхности точку планеты. Найдите расстояние, которое ему придется преодолеть при условии, что он добирается к цели кратчайшим путем.

Input

В единственной строке три целых числа A, B, C.

Output

Единственное вещественное число — расстояние, с абсолютной или относительной погрешностью не более 10^{-8}.

Limits

$1 \leqslant A, B, C \leqslant 1\,000$

Example

standard input	standard output
1 1 1	2.2360679774998

Problem K. K-edges graph (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 секунда (*1.5 секунды для Java*)
Memory limit:	512 мебибайт

Дан ациклический ориентированный граф из N вершин и K ребер. Требуется найти количество ребер в его транзитивном замыкании.

Транзитивное замыкание графа G — граф G', состоящий из множества вершин исходного графа G и множества ребер (u, v) таких, что существует путь из вершины u в вершину v в графе G.

Кнут знает, как решать эту задачу, а Вы?

Input

В первой строке два числа N и K. Далее K строк, в каждой из которых по два целых числа a_i и b_i, означающих наличие ребра, ведущего из вершины a_i в b_i. Граф не содержит петель, циклов и кратных ребер.

Output

Вывести единственное число — количество ребер в транзитивном замыкании.

Limits

$1 \leqslant N \leqslant 50\,000$

$0 \leqslant K \leqslant 50\,000$

$1 \leqslant a_i, b_i \leqslant N$

Example

standard input	standard output
5 6 1 2 2 3 3 5 4 5 1 5 1 3	7

Problem L. Lake (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 секунда
Memory limit:	256 мебибайт

Саша находится в начале координат. Но очень хочет попасть на берег озера, который почему-то находится в точке (x, y). Возможно, потому что там растет дерево. Он купил в ближайшем магазине сапоги-скороходы, с помощью которых может шагать на расстояние ровно d.

Определите, какое минимальное количество шагов потребуется, чтобы оказаться в желаемом месте. Точка (x, y) не совпадает с началом координат.

Саша может наступать в любую точку нашего плоского мира.

Input

В единственной строке три целых числа x, y, d.

Output

Единственное число — минимальное количество шагов.

Limits

$-10\,000 \leqslant x, y \leqslant 10\,000$

$1 \leqslant d \leqslant 10\,000$

Example

standard input	standard output
6 3 2	4

Problem M. Match them up (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 секунда
Memory limit:	256 мебибайт

В этой задаче вам придется искать максимальное паросочетание. Да не простое, а лексикографически минимальное.

Дан двудольный граф из N вершин в левой доле, N вершин в правой доле и K ребер между ними.

Максимальное паросочетание — максимальное по мощности подмножество ребер графа M, такое, что любая вершина графа инцидентна не более чем одному ребру из M.

Пусть множество ребер $\{(u_i, v_i)\}$ — максимальное паросочетание, где u_i — вершины из левой доли, v_i — вершины из правой доли. Выпишем все вершины в одну последовательность в порядке: u_1, v_1, u_2, v_2, ..., u_m, v_m.

Найдите максимальное паросочетание с лексикографически минимальной такой последовательностью вершин.

Input

В первой строке два числа N и K — количество вершин в каждой из долей и количество ребер. Далее K строк с описанием ребер. Каждое ребро описывается парой чисел a_i и b_i, где a_i — вершина из левой доли, а b_i — из правой доли.

Output

В первой строке выведите размер максимального паросочетания m. Далее m строк по два числа u_i, v_i в каждой, где u_i — вершина из левой доли, а v_i — вершина из правой доли, соответствующие i-ому ребру паросочетания.

Limits

$1 \leqslant N \leqslant 10^3$

$0 \leqslant K \leqslant 10^5$

$1 \leqslant a_i, b_i \leqslant N$

Example

standard input	standard output
3 5 1 2 1 3 3 2 2 3 2 1	3 1 3 2 1 3 2

Problem N. Need for sum thing (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 секунды
Memory limit:	256 мебибайт

Пусть есть треугольная таблица из N строк. Первая строка таблицы состоит из одного элемента, вторая строка — из трех, третья — из пяти и т.д. i-тая строка состоит из $2i-1$ элемента. Средние элементы всех строк образуют один столбец. Таким образом, таблица представляет собой равнобедренный прямоугольный треугольник. В верхнем элементе таблицы находится прямой угол, а в нижней строке — гипотенуза.

$$\begin{array}{ccccccccc} & & & & a_{11} & & & & \\ & & & a_{21} & a_{22} & a_{23} & & & \\ & & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & & \\ & & & & \dots & & & & \end{array}$$

Каждый элемент таблицы — некоторая цифра от '0' до '9'.

Требуется ответить на Q запросов суммы всех элементов некоторой треугольной области исходной таблицы. Каждый запрос имеет вид:

r_i, c_i, k_i

Область i-того запроса представляет собой равнобедренный прямоугольный треугольник из k_i строк с вершиной в (r_i, c_i):

$$\begin{array}{ccccccc} & & & a_{r_i,\ c_i} & & & \\ & & a_{r_i+1,\ c_i} & & a_{r_i+1,\ c_i+1} & & a_{r_i+1,\ c_i+2} \\ & & \dots & & \dots & & \dots \\ a_{r_i+k_i-1,\ c_i} & a_{r_i+k_i-1,\ c_i+1} & & \dots & & a_{r_i+k_i-1,\ c_i+2k_i-3} & a_{r_i+k_i-1,\ c_i+2k_i-2} \end{array}$$

Первая строка состоит из элемента (r_i, c_i). Вторая строка — из трех элементов: (r_i+1, c_i), (r_i+1, c_i+1), (r_i+1, c_i+2), третья из пяти и т.д.

Нужно найти сумму всех элементов треугольника из запроса.

Поскольку запросов много, они генерируются программно:

$A_1 = 1$

$A_i = (1\,234\,567 \cdot A_{i-1} + 7\,654\,321) \mod 1\,000\,000\,007$, при $2 \leqslant i \leqslant Q$

$r_i = A_i \mod N + 1$

$c_i = A_i \mod (2r_i - 1) + 1$

$k_i = A_i \mod (N - r_i + 1) + 1$

Input

В первой строке N и Q — количество строк исходного треугольника и количество запросов. Далее в N строках идет описание таблицы. В i-той из них ровно $2i-1$ цифр — элементы таблицы (между цифрами нет пробелов).

Output

Поскольку запросов много, выведите одно число — сумму всех ответов на запросы.

Limits

$1 \leqslant N \leqslant 10^3$

$0 \leqslant Q \leqslant 5 \cdot 10^6$

Example

standard input	standard output
3 5 1 234 56789	42

Explanation

Исходный треугольник:
1
234
56789

Первый запрос:
3
678, сумма 24.

Второй запрос: 8, сумма 8.

Третий запрос: 6, сумма 6.

Четвертый запрос: 1, сумма 1.

Пятый запрос: 3, сумма 3.

Problem O. Open air (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 секунда
Memory limit:	256 мебибайт

Есть полный набор доминошек, на каждой из которых записано по два числа от 0 до N (N — четное). Причем каждая пара чисел $\{a, b\}$ встречается ровно один раз. Нетрудно посчитать, что количество доминошек в наборе — $\frac{(N+1)(N+2)}{2}$.

Требуется расположить все доминошки горизонтально в $N+1$ ряд так, чтобы суммы чисел, записанных на всех доминошках в каждом ряду, были равны. Гарантируется, что это всегда можно сделать.

Input

В единственной строке четное число N.

Output

Вывести $N+1$ строку по $N+2$ числа в каждой: $a_1, b_1, a_2, b_2, \ldots, a_k, b_k$, $k = \frac{N+2}{2}$, где каждая пара a_i, b_i соответствует очередной доминошке. Каждую доминошку нужно использовать ровно один раз. Пару чисел на каждой доминошке можно выводить в любом порядке. Если решений несколько, можете вывести любое из них.

Limits

$2 \leqslant N \leqslant 100$, N — четное

Example

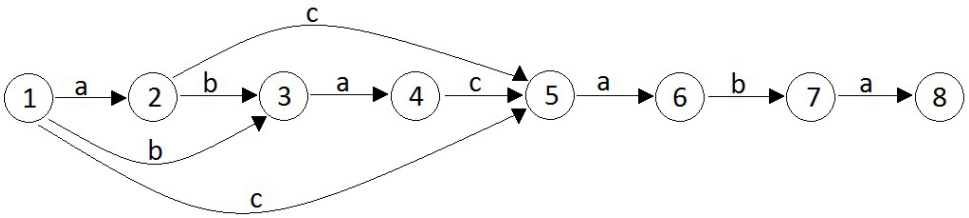
standard input	standard output
2	1 2 0 1 1 1 0 2 0 0 2 2

Problem P. Pseudo automaton (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 секунда
Memory limit:	256 мебибайт

Дана строка S. Построить детерминированный конечный автомат, принимающий все суффиксы строки S (и возможно другие конечные строки). Автомат должен состоять из минимального числа состояний. Каждое состояние автомата объявляется финальным. Начальное состояние имеет номер 1.

На изображении ниже показан автомат из тестового примера для строки "`abacaba`"



Input

В единственной строке слово S, состоящее из строчных латинских букв.

Output

В первой строке два числа N и K — количество состояний и количество переходов. Далее K строк, в каждой из которых по два числа a_i, b_i и буква c_i, означающие наличие перехода из состояния a_i в b_i по букве c_i. Переходы можно выводить в любом порядке. Если решений несколько, можете вывести любое из них.

Limits

$1 \leqslant |S| \leqslant 5\,000$

$1 \leqslant a_i, b_i \leqslant N$

'a' $\leqslant c_i \leqslant$ 'z'

Example

standard input	standard output
abacaba	8 10 1 2 a 1 3 b 1 5 c 2 3 b 2 5 c 3 4 a 4 5 c 5 6 a 6 7 b 7 8 a

Problem Q. Quiz (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 секунда
Memory limit:	256 мебибайт

Каждый из вас скорее всего знаком с детской игрой "пятнашки". В этой задаче требуется найти решение для некоторой позиции.

Игра заключается в следующем: есть квадратное поле $N \times N$, разбитое на клетки 1×1. В во всех клетках кроме одной есть фишки, на каждой из которых записано число от 1 до $N^2 - 1$. Каждое число встречается ровно один раз. На этом поле можно осуществлять ходы фишками, а именно, за один ход можно передвинуть одну из соседних с пустым местом фишек на это пустое место. При этом на месте фишки образуется пустое место. Фишки не могут покидать пределы поля.

Решить головоломку — значит расположить фишки в определенном порядке:

$$\begin{array}{cccc} 1 & 2 & \dots & N-1 & N \\ N+1 & N+2 & \dots & 2N-1 & 2N \\ \vdots & & & & \vdots \\ N^2-N+1 & N^2-N+2 & \dots & N^2-1 & \end{array}$$

Пустое место должно оказаться в последней клетке последней строки.

Для обычных пятнашек это выглядит как:

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & \end{array}$$

Дана позиция, найти последовательность ходов (не обязательно кратчайшую, возможно пустую), которая ее решает. Либо сказать, что позиция не имеет решения.

Input

В первой строке число N — размер поля. Далее N строк по N чисел в каждой. Числа от 1 до N^2-1 соответствуют фишкам, 0 соответствует пустому месту. Каждое число от 0 до $N^2 - 1$ встречается ровно один раз.

Output

Если позиция не имеет решения, вывести "`No`". В противном случая в первой строке вывести "`Yes`", а во второй — строку из ходов (без пробелов):

'`L`' означает, что на пустое место надо передвинуть фишку, находящуюся слева от него.

'`R`' означает, что на пустое место надо передвинуть фишку, находящуюся справа от него.

'`U`' означает, что на пустое место надо передвинуть фишку, находящуюся сверху от него.

'`D`' означает, что на пустое место надо передвинуть фишку, находящуюся снизу от него.

Количество ходов не должно превышать 2 500 000. Если решений несколько, можете вывести любое из них.

Limits

$2 \leqslant N \leqslant 50$

$0 \leqslant a_{ij} \leqslant N^2 - 1$

Example

standard input	standard output
2 0 3 2 1	Yes DRULDR
2 2 1 3 0	No

Problem R. Reduction (Division 2 Only!)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 секунда
Memory limit:	256 мебибайт

Дана строка S и список M, состоящий из N слов, каждое из которых длины L. За одну операцию можно выбрать некоторую подстроку строки S, если такая встречается в качестве слова в списке M, и вырезать из строки S. После чего оставшиеся части строки S, если такие есть, склеиваются. Определить, за какое минимальное количество операций можно уничтожить всю строку S. Гарантируется, что это можно сделать.

Input

В первой строке слово S. Во второй строке целое число N — количество слов в списке. Далее N строк, в каждой из которых по одному слову из списка M. Все слова состоят только из строчных латинских букв.

Output

Одно число — минимальное количество операций, необходимое для уничтожения строки S.

Limits

$1 \leqslant |S| \leqslant 100$

$1 \leqslant N \leqslant 100$

$1 \leqslant |M_i| \leqslant 100$

$1 \leqslant L \leqslant |S|$

Example

standard input	standard output
abacabada 4 aba aca ada abb	3