

Problem A. Balance

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Назовём матрицу A размерности $N \times N$ *сбалансированной*, если $A[i][j] + A[i+1][j+1] = A[i+1][j] + A[i][j+1]$ для всех $1 \leq i, j \leq N-1$.

Дана матрица A размерности $N \times N$. Ваша задача — вывести другую матрицу B той же самой размерности так, что матрица B сбалансирована и $B[i][j] \geq A[i][j]$ для всех $1 \leq i, j \leq N$.

При этом матрица B должна иметь наименьшую возможную сумму элементов.

Input

Первая строка входа содержит целое число N , количество строк и столбцов в матрице ($1 \leq N \leq 50$).

Далее задана матрица A — каждая из последующих N строк содержит по N целых чисел. Гарантируется, что $0 \leq A[i][j] \leq 35\,000$ для всех $1 \leq i, j \leq N$.

Output

В первой строке выведите сумму элементов найденной Вами матрицы B .

В последующих N строках выведите саму матрицу в том же формате, в котором она была задана на входе.

Решением считается любая матрица, соответствующая требованию задачи и имеющая минимальную сумму. Отметим, что ограничение в 35 000 на матрицу B не распространяется.

Example

standard input	standard output
4	16
1 1 1 1	1 1 1 1
1 1 1 1	1 1 1 1
1 1 1 0	1 1 1 1
1 1 1 1	1 1 1 1

Problem C. Gravity

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

Вам задана матрица $N \times M$, заполненная символами “.” (свободное пространство) и “#” — часть объекта. Каждая максимальная четырёхсвязная компонента, составленная из символов “#” задаёт один цельный объект.

Во время действия задачи, объекты остаются цельными и не могут склеиться или разделиться каким-либо образом. Все части одного объекта двигаются одинаково.

Объекты начинают падать вниз (к последней строке матрицы) с постоянной скоростью. Объекты падают строго параллельно, безо всякого вращения.

Каждую секунду все объекты пытаются сдвинуться вниз. Если движение приводит к тому, что объект пересекает нижнюю границу матрицы или перекрывается с другим объектом, объект не сдвигается и прекращает падение, в противном случае в эту секунду он сдвигается на один ряд вниз

По заданному изначальному расположению объектов вычислите ситуацию в момент, когда все объекты прекратят падение.

Input

Первая строка входных данных содержит два целых числа N и M ($1 \leq N, M \leq 2000$).

Последующие N строк задают матрицу. Каждая из этих строк содержит по M символов, каждый из которых может быть либо “.”, либо “#”.

При этом символы в одной строке **не разделены** пробелами.

Output

Выведите матрицу, задающую расположение объектов после того, как все объекты упадут, в формате, соответствующем формату ввода (за исключением того, что размерность матриц не выводится).

Example

standard input	standard output
10 10
.....
..#####..	..#####..
..#...#..	..#...#..
..#.#.#..	..#...#..
..#.#.#..	..#...#..
..#...#..	..#...#..
..#####..	..#####..
.....#..
..#...#..	..#...#..
.....#..	

Problem D. Infinite Pattern Matching

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Рассмотрим бесконечную двоичную строку I , построенную конкатенацией двоичных представлений всех целых положительных чисел, перечисленных по возрастанию: $I = "11011100\dots"$.

Вам задана двоичная строка A . Ваша задача — найти наименьшее целое число L такое, что A — суффикс строки $I[1 \dots L]$.

Input

Входной файл содержит двоичную строку A , $1 \leq |A| \leq 55$.

Output

Выведите одно целое число L — ответ к задаче.

Examples

standard input	standard output
11	2
011011	42

Problem I. Taxi

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

Дано неориентированное дерево из N вершин. Длина каждого ребра — целое положительное число. На дереве находятся M такси и M пассажиров, каждое такси и каждый пассажир находятся в некоторой вершине дерева (при этом в одной вершине может быть как несколько такси, так и несколько пассажиров, а также несколько пассажиров и несколько такси одновременно).

Приложение для вызова такси назначает каждому пассажиру такси. Так как по правилам соответствующей компании пассажир платит за расстояние, которое такси проехало перед прибытием к нему, приложение стремится максимизировать прибыль компании, максимизировав суммарное расстояние, которые такси проедут до того, как забрать пассажиров. При этом каждому пассажиру должно приехать одно такси и каждое такси забирает ровно одного пассажира.

Всего есть N^{2M} различных способов расстановки пассажиров и такси на дереве. Для каждого из этих способов мы рассчитаем суммарное максимальное расстояние в соответствии с алгоритмом приложения. Требуется вычислить сумму этих расстояний по модулю $10^9 + 7$.

Input

Первая строка входных данных содержит два целых числа N и M ($1 \leq N, M \leq 2500$).

Каждая из последующих $N - 1$ строк содержит по три целых числа x , y и l . Это обозначает, что вершины x и y соединены ребром длины l ($1 \leq l \leq 10\,000$).

Гарантируется, что заданный граф является деревом.

Output

Выведите одно число — ответ к задаче по модулю $10^9 + 7$.

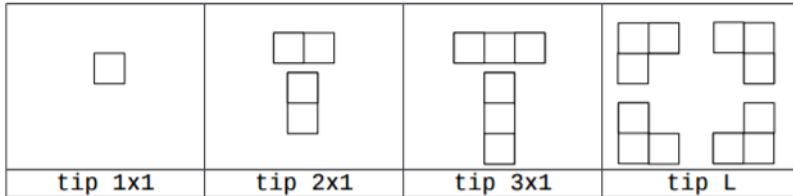
Example

standard input	standard output
5 2	10784056
4 5 9805	
3 4 2001	
2 3 6438	
1 3 3790	

Problem J. Tris

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

У вас есть некоторое количество полимино. Возможны следующие типы полимино:



Вы можете вращать каждое полимино и симметрично отражать его. Ваша задача — разместить все полимино на прямоугольной доске размерности не более, чем 800 по каждому измерению так, чтобы получился один несамокасающийся цикл.

Более формально:

- Все полимино должны полностью находиться на доске, квадраты полимино должны совпадать с полями доски.
- Никакие два полимино не могут перекрываться.
- Если какое-то поле занято полимино, то ровно два из четырёх соседних полей могут быть заняты.
- Все занятые поля 4-связны, то есть вы можете пройти между любыми двумя занятыми полями только по занятым клеткам, переходя в поля, имеющим общую сторону.
- Внутренность цикла должна быть одной 4-связной областью.

Input

Входные данные состоят из одной строки, содержащей четыре целых числа — количество полимино каждого типа в порядке, в котором они изображены на иллюстрации.

Гарантируется, что каждое число равно как минимум 2 и как максимум 100, и что ответ для каждого набора данных существует.

Output

В первой строке выведите два целых числа N и M ($N, M \leq 800$), обозначающие количество строк и столбцов доски. Последующие строки должны описывать расстановку полимино в следующем формате:

- Расстановка должна содержать целые числа от 0 до общего количества полимино, включительно.
- Полям, занятым одним полимино, должно соответствовать одно и то же значение.
- Полям, занятым разным полимино, должны соответствовать разные значения.
- Полям, которые не заняты ни одним полимино, должно соответствовать значение 0.

Если ответов несколько, разрешается вывести любой из них.

Example

standard input	standard output	picture
3 4 3 4	11 6 0 1 2 4 4 4 1 1 0 0 0 3 8 0 0 0 3 3 8 0 0 0 9 0 8 0 0 0 9 9 10 0 0 0 0 13 10 0 0 0 0 11 12 0 0 0 0 11 12 0 0 0 0 14 6 0 0 0 0 7 6 5 5 5 7 7	

Problem M. Bus

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 N. Ancestry

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

При определении порядка наследования в Байтландии 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 Байтландии 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 O. Olympiad

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

Министерство образования Бйатландии 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 Бйатландии.

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 Бйатландии, 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

Problem P. Anagrams

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

Два слова являются анаграммами, если они состоят из одних и тех же букв, заданных в различном порядке. Например, слова `word` и `drow` анаграммами являются, а `word` и `worm` — нет. В этой задаче вам будет дан набор попарно различных слов. Вам необходимо найти слово, для которого в наборе количество анаграмм максимально и вывести как это слово, так и количество его анаграмм в оставшейся части набора.

Input

В первой строке входного файла задано одно целое число n — количество слов в наборе ($0 \leq n \leq 1000$). Далее следуют n слов, каждое с отдельной строки. Все слова состоят из строчных латинских букв, длина каждого слова не превышает 6. Гарантируется, что в наборе будет содержаться как минимум два слова, являющиеся анаграммами друг друга.

Output

В выходной файл выведите первое (в порядке следования во входном файле) слово, для которого в наборе количество анаграмм максимально, а после этого слова через пробел количество анаграмм этого слова (исключая его само), присутствующих в наборе.

Example

standard input	standard output
6 nat cat act out tac ant	cat 2
6 worm word galo drown goal drow	word 1

Problem Q. Banking

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

Крупный коммерческий банк «Шарабанк» в течение ряда лет собирал статистику ежедневного роста и падения акций банка. Основываясь на этих числах, владельцы банка решили вычислить «оптимальную стратегию» для акционера — в какой из дней было наиболее выгодно всего купить акции банка, а в какой — продать.

Ваша задача — по заданной последовательности чисел (ежедневным изменениям стоимости одной акции) найти «оптимальную тактику» при условии, что можно один раз купить акции банка и один раз полностью продать все купленные акции. Докупать акции или продавать акции частично не разрешается.

Input

Входной файл состоит из двух строк. Первая строка содержит целое число N ($1 \leq N \leq 10^6$) — время, в течение которого собиралась статистика. Вторая строка содержит N целых чисел p_i ($-1000 \leq p_i \leq 1000$) — изменение стоимости акции в i -й день. Гарантируется, что хотя бы одно из чисел p_i положительно.

Output

В выходной файл выведите два целых числа k и l ($1 \leq k \leq l \leq N$) — номер дня (нумерация начинается с единицы), когда лучше всего будет купить и продать акции соответственно. В случае, если решение не единственно, выберите решение с наименьшим k , а при совпадении k — с наименьшим l .

Example

standard input	standard output
11 -3 1 -1 2 3 1 -1 2 -3 -5 7	2 8
9 1 -2 3 -1 -1 3 -2 2 -4	3 6

Problem R. Dimensions

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

На уроке математики в начальных классах ученикам была роздана таблица, в каждой строке которой были записаны высота, ширина, длина и объём некоторого прямоугольного параллелепипеда. При этом ровно одно значение в каждой строке было зачёркнуто. Ученики должны были восстановить исходный вид таблицы. Напишите программу, которая решает аналогичную задачу.

Input

Входной файл состоит из нескольких тестовых примеров, каждый из которых состоит из одной строки, содержащей 4 целых числа w , h , l and v — ширину, высоту, длину и объём прямоугольного параллелепипеда соответственно ($0 < l, w, h < 100, 0 < v < 10^5$). Зачёркнутые числа представлены нулями, то есть в каждой строке одно из чисел заменено нулём. Входной файл завершается строкой из четырёх нулей, которая не должна обрабатываться.

Output

Для каждого тестового примера в отдельной строке выведите 4 целых числа — ширину, высоту, длину и объём соответствующего параллелепипеда, заменив во входном примере 0 на вычисленное значение соответствующего параметра. Гарантируется, что вычисленное значение всегда будет целым.

Example

standard input	standard output
1 0 2 6	1 3 2 6
5 5 5 0	5 5 5 125
0 2 2 80	20 2 2 80
8 0 9 576	8 9 9 576
0 0 0 0	