

Problem A. The Motorway

Input file: **standard input**
Output file: **standard output**
Time limit: **9 seconds**
Memory limit: **128 mebibytes**

Компания Autobyte участвует в эксплуатации одной из платных дорог в Байтеландии. До недавнего времени оплата взималась только при въезде на дорогу и не зависела от расстояния, которое проехал водитель. Сейчас компания планирует построить дополнительно несколько пунктов взимания оплаты, распределённых по всей длине дороги.

На дорогу можно въехать с n боковых дорог (точка въезда задаётся её расстоянием от начала дороги). Компания собирается $n + 1$ пунктов взимания оплаты таким образом, чтобы расстояние между любыми двумя последовательными пунктами взимания оплаты было одинаково, при этом между ними должна быть точка въезда. Также между любыми двумя точками въезда должен быть пункт взимания оплаты. Выяснилось, что такая расстановка возможна.

Требуется вычислить наибольшее и наименьшее расстояния между пунктами взимания оплаты. Более формально, требуется найти наибольшее и наименьшее значение l , для которого существует точка размещения первого пункта взимания оплаты b_0 такая, что пункты оплаты будут располагаться в точках $b_0 + l, b_0 + 2l, \dots, b_0 + nl$.

В случае, если пункт оплаты совпадает с точкой въезда, он может быть учтён «в обе стороны», иначе говоря, j -я точка въезда должна быть размещена в интервале $[b_0 + (j - 1)l, b_0 + jl]$.

Input

Первая строка входного файла содержит одно целое число n ($3 \leq n \leq 1\,000\,000$): — количество точек въезда. Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — расстояние от начала дороги до точек въезда в порядке их следования.

Output

Выведите два вещественных числа — наибольшее и наименьшее возможное расстояние между двумя соседними пунктами взимания оплаты. Гарантируется, что расстояние между ними не менее 10^{-9} .

Решение будет признано правильным, если оно лежит в интервале $[x(1 - \epsilon) - \epsilon, x(1 + \epsilon) + \epsilon]$, где x — правильный ответ, $\epsilon = 10^{-8}$.

Examples

standard input	standard output
6 2 3 4 5 6 7	0.833333333333 1.250000000000

Problem B. Bytehattan (Division 1 Only!)

Input file: standard input
Output file: standard output
Time limit: 15 seconds
Memory limit: 128 mebibytes

Крупнейший торговый и промышленный город Байтландии расположен на нескольких островах. Самый известный из них — Байтхэттен. Автомобильные пробки на Байтхэттене, бывает, блокируют движение по целой улице. Байтазару поручено отслеживать интенсивность движения на Байтхэттене.

Улицы Байтхэттена формируют регулярную решётку $n \times n$. Представим карту Байтхэттена в виде координатной плоскости: для каждой пары целых чисел x, y , такой, что $1 \leq x, y \leq n$, в точке с координатами (x, y) находится перекрёсток. Любые два перекрёстка на расстоянии 1 соединены улицей длины 1.

Байтазар получает сообщения о перекрытии улиц «по факту перекрытия». После получения соответствующей информации Байтазар должен определить, возможно ли проехать между перекрёстками, соединёнными этой улицей, по улицам, которые в данный момент не перекрыты.

Напишите программу, которая отвечает на подобные запросы.

Input

Первая строка входа содержит два целых числа n и k ($2 \leq n \leq 1500$, $1 \leq k \leq 2n(n-1)$) — количество улиц Байтхэттена и количество сообщений о блокировании некоторой улицы. Каждая из последующих k строк содержит информацию насчёт блокирования одной из улиц. Сообщения о блокировании идут в хронологическом порядке. Каждое из таких сообщений задаёт две улицы. При этом ровно одна из этих улиц блокируется¹, причём заблокированная улица определяется следующим способом. Если для предыдущего запроса был получен положительный ответ, то блокируется первая из указанных улиц, если нет — то вторая. При этом для самого первого запроса блокируется первая улица. Каждая улица может быть заблокирована один раз и не может быть разблокирована.

Улицы задаются парой целых чисел a_i, b_i ($1 \leq a_i, b_i \leq n$), за которой следует буква c_i ($c_i \in \{N, E\}$). Данная тройка задаёт улицу, начинающуюся с перекрёстка (a_i, b_i) . В случае, если $c_i = N$, другой конец улицы имеет координаты $(a_i, b_i + 1)$. В случае, если $c_i = E$, другой конец улицы имеет координаты $(a_i + 1, b_i)$. Если $c_i = N$, то $b_i < n$, аналогично, если $c_i = E$, то $a_i < n$.

Output

Выведите ровно k строк. i -я из этих строк должна содержать ответ на соответствующий запрос: если объезд заблокированной в i -м сообщении дороги возможен, выведите “ТАК” («да» по-польски), иначе выведите “NIE” («нет» по-польски).

Examples

standard input	standard output
3 4	TAK
2 1 E 1 2 N	TAK
2 1 N 1 1 N	NIE
3 1 N 2 1 N	NIE
2 2 N 1 1 N	

¹ Данный формат ввода используется жюри для того, чтобы каждое перекрытие улицы было обработано перед обработкой последующих перекрытий.

Problem C. The Carpenter (Division 1 Only!)

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 128 mebibytes

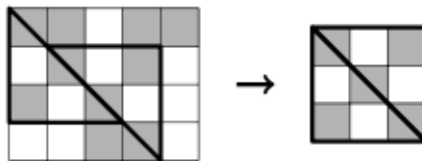
Байтазар собрался поиграть в шашки, однако шахматной доски поблизости не обнаружилось. Правда, нашлась деревянная доска размера $n \times m$, разделённая на nm одинаковых квадратов. Каждый квадрат покрашен или в чёрный, или в белый цвет (при этом цвета не обязаны идти в шахматном порядке).

Байтазар решил распилить доску таким образом, чтобы, используя некоторые из получившихся кусков, можно было составить квадратную шахматную доску требуемого размера, в которой любые две соседние клетки имели различные цвета.

Так как вырезать соответствующий квадратный кусок можно не всегда, Байтазар решил вырезать два треугольных куска, чтобы склеить их вместе и сделать доску. Отпиленные части доски можно двигать и поворачивать произвольным образом.

По заданной исходной доске вычислите максимальную сторону корректной квадратной шахматной доски, которая может быть получена вышеописанным способом.

На иллюстрации показано, как из заготовки размера 4×5 и двух треугольников делается корректная шахматная доска 3×3 :



Input

Первая строка входа содержит два целых числа n и m ($1 \leq n, m \leq 1000$) — размеры заготовки. Каждая из последующих n строк содержит по m целых чисел: j -е число в j -й строке ($1 \leq i \leq n$, $1 \leq j \leq m$) задаёт цвет поля, находящегося на пересечении j -го столбца и i -й строки заготовки. 0 соответствует белому полю, 1 — чёрному.

Output

Выведите одно целое число — максимальную длину стороны корректной шахматной доски, которую можно получить описанным в задаче способом.

Examples

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

Problem D. Demonstrations

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 128 mebibytes

В ближайшее воскресенье отмечается национальный праздник Байтландии — День Бита.

Однако байтландское общество расколото по ключевому вопросу — каково оптимальное количество бит в байте. Количество партий, отстаивающих разные точки зрения, весьма велико, и каждая партия планирует организовать митинг в поддержку восстановления «исторической справедливости».

Сотрудники столичной полиции недовольны ситуацией, при которой им придётся весь праздник провести на работе, охраняя митингующих. При этом, пока в городе идёт хотя бы один митинг, всем полицейским выплачиваются сверхурочные в размере одного байтландского тугрика в минуту на одного полицейского.

В целях экономии бюджета мэр столицы принял решение запретить не более двух митингов. Митинги, которые попадают под запрет, должны быть выбраны таким образом, что суммарное время, в течение которого придётся платить сверхурочные полицейским, будет минимально.

Помогите мэру определить, сколько тугриков в расчёте на одного полицейского он может сэкономить

Input

Первая строка входа содержит одно целое число n ($2 \leq n \leq 500\,000$): количество запланированных митингов. Последующие n строк описывает один митинг: i -я из этих строк содержит два целых числа a_i и b_i ($0 \leq a_i < b_i \leq 10^9$), — время начала и конца i -го митинга в минутах, соответственно.

Output

Выведите одно целое неотрицательное число — максимальную сумму сверхурочных в расчёте на одного полицейского, которую можно сэкономить.

Examples

standard input	standard output
5 0 9 1 4 2 5 7 9 6 7	4

Problem E. The Exam

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 128 mebibytes

Профессор Байтони готовит вариант для письменного экзамена. Он уже подготовил n вопросов. Вопросы ранжированы в порядке сложности и занумерованы последовательными натуральными числами от 1 до n .

Сейчас профессор обдумывает последовательность проведения экзамена. Ему интересно, смогут ли студенты самостоятельно заметить разницу в сложности вопросов. Для этого он планирует расположить вопросы в таком порядке, чтобы у любых двух подряд идущих вопросов номера различались как минимум на k .

Помогите профессору найти соответствующую последовательность.

Input

Вход содержит два целых числа n и k ($2 \leq n \leq 1\,000\,000$, $1 \leq k \leq n$): количество вопросов и требуемое минимальное расстояние между номерами.

Output

Выведите одну строку, содержащую требуемую последовательность номеров вопросов (иначе говоря, перестановку первых n натуральных чисел такую, что любые два соседних числа различаются не менее, чем на k). Если таких последовательностей несколько, выведите любую. Если такой последовательности не существует, выведите слово “NIE” (*нет* по-польски).

Examples

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

Problem F. Speed Cameras

Input file: `standard input`
Output file: `standard output`
Time limit: 5 seconds
Memory limit: 256 mebibytes

Мэр столицы Байтландии планирует разместить в городе некоторое количество камер контроля скорости движения автомобилей. В столице n перекрёстков, занумерованных от 1 до n , и $n - 1$ отрезков улиц, на которых введено двустороннее движение. Каждый из этих отрезков соединяет два перекрёстка, никакие два отрезка не имеют общих точек иначе, чем в начале или конце. Между любыми двумя перекрёстками можно проехать по сети улиц.

Камеры должны быть размещены на перекрёстках (не более одной на перекрёстке), при этом на любом маршруте между двумя перекрёстками по улицам столицы, который не содержит один и тот же перекрёсток дважды, встретится не более k камер (начальный и конечный перекрёстки учитываются), а количество камер с учётом вышеуказанных ограничений должно быть максимально.

Требуется определить, на каких перекрёстках должны быть расставлены камеры.

Input

Первая строка содержит два целых числа n и k ($1 \leq n, k \leq 1\,000\,000$): количество перекрёстков в столице и максимальное количество камер на маршруте. Далее следуют $n - 1$ строк, задающих отрезки улиц: i -я из этих строк содержит два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$), обозначающие, что перекрёстки a_i и b_i соединены отрезком улицы с двусторонним движением.

Output

В первой строке выведите одно целое число m : максимальное количество камер, которые могут быть установлены. Вторая строка должна содержать любую из корректных расстановок m камер, соответствующую условию задачи.

Examples

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

Problem G. Game (Division 1 Only!)

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 128 mebibytes

Двое играют в следующую игру. В урне имеется чётное количество шариков, на каждом шарике написана ровно одна цифра. Игроки по очереди случайным образом вытягивают шары из каждой урны. Игра заканчивается, когда урна становится пустой. Игрок, у которого произведение вытягиваемых цифр максимально, выигрывает.

Требуется по заданному набору шариков в урне определить, может ли игра завершиться вничью.

Input

В первой строке задано целое число t ($1 \leq t \leq 1000$) — количество тестовых примеров.

Каждая из последующих t строк содержит 10 неотрицательных целых чисел k_0, \dots, k_9 ($0 \leq k_i \leq 10^{15}$), где k_i обозначает количество шариков, на которых написана цифра i . Гарантируется, что сумма чисел k_i является неотрицательным чётным числом.

Output

Для каждого тестового примера выведите “TAK” (*да* по-польски) в случае, если ничья возможна, и “NIE” (*нет* по-польски) в противном случае.

Examples

standard input
5 0 1 0 1 1 4 1 0 5 1 0 1 1 0 3 0 0 0 0 3 1 1 0 4 0 0 2 0 0 2 1000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000 0 999999 999999 1000000 1000000 1000000 1000000 1000000 1000000
standard output
TAK NIE NIE TAK NIE

Problem H. The Hero (Division 1 Only!)

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

Известный древнебайтландский герой после очередной великой победы собрался в обратный путь на остров Битака. Однако герою известно, что боги подготовили на пути назад разные ловушки...

Всего в море n островов, занумерованных последовательными натуральными числами от 1 до n . Победа была одержана на острове с номером 1, Битака имеет номер n . Некоторые из островов соединены *однонаправленными* морскими маршрутами, при этом ни от одного острова не отходит более 10 маршрутов. Занумеруем эти маршруты последовательными натуральными числами от 1 до m ; i -й маршрут ведёт с острова a_i на остров b_i , и для его прохождения паруснику требуется d_i дней: если корабль отплыл по i -му маршруту с острова a_i , стартуя с него на рассвете i -го дня, то он достигнет острова b_i на рассвете $j + d_i$ -го дня. Менять маршрут (в частности, обойти остров назначения) и задерживаться в открытом море корабль не может. Около какого-либо острова корабль может задерживаться на произвольное время. Герой начинает путь на Битаку не ранее утра первого дня.

Боги установили на пути героя p ловушек. Каждая из ловушек находится на некотором острове w_i и действует, начиная с дня s_i по день k_i включительно. Если корабль находится около острова, на котором находится действующая ловушка, он гибнет вместе со всем экипажем. Гарантируется, что на Битаке ловушек нет и что в первый день ловушек нет на острове, на котором герой одержал победу.

Герой хочет запланировать обратный путь так, чтобы избежать всех ловушек и за минимальное количество дней вернуться на Битаку.

Input

Первая строка содержит два целых числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 1\,000\,000$): количество островов и количество морских маршрутов, соответственно. Последующие m строк описывают морские пути: i -я из них содержит три целых числа a_i , b_i , d_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq d_i \leq 10^9$), обозначающие, что i -й морской путь ведёт от острова a_i на остров b_i и занимает d_i дней. Все пути являются однонаправленными. Каждый остров является отправной точкой не более, чем для 10 маршрутов.

Следующая строка содержит целое число p ($0 \leq p \leq 100\,000$), задающее количество ловушек. Следующие p строк содержат описания ловушек: i -я из них содержит три целых числа w_i , s_i , k_i ($1 \leq w_i < n$, $1 \leq s_i \leq k_i \leq 10^9$), обозначающих, что i -я ловушка находится на острове w_i и активна, начиная с дня s_i по день k_i включительно. Если $w_i = 1$, то $s_i > 1$.

Output

В случае, если прибыть на Битаку, избегнув ловушек, невозможно, выведите слов "NIE" (*нет* по-польски). Иначе выведите целое число d — минимальное число дней, которое потребуется герою для путешествия (то есть на рассвете $d + 1$ -го дня корабль должен находиться около Битаки).

Examples

standard input	standard output
5 6 1 2 3 1 4 13 2 3 1 2 4 2 3 2 2 4 5 1 5 1 2 4 1 8 8 2 6 7 2 10 11 4 6 7	10

Note

В примере из условия герой отплывает с острова 1 на рассвете первого дня, на четвёртый он прибывает к острову 2, ждёт один день, отправляется на остров 3. Там он оказывается на шестой день, после чего тут же идёт назад на остров 2, после чего он на восьмой день отправляется к острову 4. На десятый день он прибывает к острову 4 и тут же отправляется на Битаку, прибывая туда на двенадцатый день.

Problem I. Genetic Engineering

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 128 mebibytes

Байтландские палеонтологи недавно нашли несколько кусков янтаря, внутри которых обнаружались древние комары. После изучения образцов находка была датирована Юрским периодом, тем самым, вполне возможно, что комары имели контакт с крупными рептилиями, которые населяли в те годы территорию нынешней Байтландии. У генетиков зародился дерзкий план: восстановить из образцов генетический материал байтозавра — наиболее распространённого на соответствующей территории ящера.

Геном байтозавра является цепочкой, составленной из аминокислот. Для упрощения мы обозначим типы аминокислот натуральными числами. Геном обладает следующим свойством: каждая аминокислота повторяется k раз подряд (тем самым длина корректной цепочки обязана делиться на k). Если мы разобьём геном на блоки, состоящие из последовательно идущих k аминокислот, каждый такой блок будет состоять из аминокислот одного и того же типа.

Генетики выделили цепочку аминокислот длины n , которая, предположительно, содержит геном байтозавра. При этом возможно, что в цепочку «встроились» внешние аминокислоты, не относящиеся к геному. Сейчас генетики хотят проверить свою гипотезу, убрав минимальное количество аминокислот из цепочки таким образом, чтобы получился корректный геном. В случае, если получается несколько корректных геномов, учёных интересует лексикографически наименьший.²

Помогите палеонтологам в решении их задачи.

Input

Первая строка ввода содержит два целых числа n и k ($1 \leq n \leq 1\,000\,000$, $2 \leq k \leq 1\,000\,000$): длину выделенной генетиками цепочки аминокислот и значение k для корректного генома. Вторая строка содержит последовательность из n целых чисел g_1, \dots, g_n ($1 \leq g_i \leq 1\,000\,000$) — типы аминокислот в цепочке.

Output

Выведите две строки. Первая строка должна содержать одно целое число m ($0 \leq m \leq n$) — длину максимально возможного корректного генома, который может получиться из заданной цепочки удалением некоторого количества аминокислот.

Вторая строка должна содержать последовательность из m чисел — типы аминокислот в корректном геноме. Если есть несколько решений, выведите лексикографически наименьшее. Если длина генома равна нулю (то есть корректную цепочку выделить не удаётся), вторая строка должна быть пустой.

Examples

standard input	standard output
16 3	9
3 2 3 1 3 1 1 2 4 2 1 1 2 2 2 2	1 1 1 2 2 2 2 2 2

²Пусть l_1 и l_2 — две цепочки аминокислот одинаковой длины. Чтобы определить, какая из них является лексикографически меньшей, требуется найти первую от начала цепочки аминокислоту, по которой эти цепочки различаются. Та цепочка, у которой на соответствующем месте находится аминокислота с меньшим номером, и является лексикографически меньшей.

Problem J. Robin Hood

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **128 mebibytes**

Как известно, знаменитый разбойник Робин Гуд грабил богатых и раздавал награбленное бедным. Однажды Робин и его шайка ограбили караван, который вёз золото в графский замок. Добычей шайки стаи n корзин. В пещере, служившей разбойникам пристанищем, они осмотрели добычу и выяснили, что i -я корзина (для $i = 1, 2, \dots, n$) содержит ровно i сумок с золотом.

Когда бедняк приходит просить у Робина несколько золотых дукатов, Робин действует следующим образом. Сначала он выбирает непустую корзину, в которой содержится наименьшее количество сумок с золотом.

В случае, если в корзине осталась только одна сумка с золотом, Робин отдаёт её бедняку, и тот, облагодетельствованный, уходит восвояси.

В случае, если в корзине осталось нечётное количество сумок с золотом (больше единицы), Робин Гуд пересыпает содержимое этой сумки в свой кошелёк, после чего начинает процедуру заново.

В случае, если в корзине осталось чётное количество сумок с золотом, Робин Гуд делит содержимое этой корзины пополам (половину оставляет в этой корзине, половину перекладывает в новую; количество пустых корзин в пещере таково, что это действие всегда возможно), после чего начинает процедуру заново.

Таким образом, в результате одной или нескольких итераций бедняк получает свою сумку с золотом — в случае, если у разбойников осталась хотя бы одна непустая корзина. Бедняки идут друг за другом до тех пор, пока корзины с золотом не опустеют.

Узнав о хитрой процедуре, остальные члены шайки обеспокоились, насколько соответствует поведение Робина правилам, принятым среди благородных разбойников. Для начала они хотят узнать, содержимое какого количества сумок перекочевало в кошелёк к Робину.

Input

Входной файл содержит одно целое число n ($1 \leq n \leq 10^9$) — количество захваченных разбойниками корзин с деньгами.

Output

Выведите одно целое число — количество сумок с деньгами, содержимое которых оказалось в кошельке у Робин Гуда к моменту, когда все корзины опустели.

Examples

standard input	standard output
5	2

Problem K. Blanket

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 128 mebibytes

Прошедшее лето в Байтландии было жарким, и жители столицы потянулись на пляжи Байтбургского водохранилища загорать. Чтобы упорядочить наплыв отдыхающих, на общественных пляжах были приняты следующие правила: пляжные коврики должны иметь размер $a \times b$ и располагаться длинной стороной перпендикулярно прямолинейному берегу водохранилища.

Проведя несколько дней на пляже, известный байтландский учёный профессор Байтони сделал следующее наблюдение: каждый отдыхающий всё время размещает свой коврик на одном и том же месте, и, хотя люди приходят и уходят на пляж в разное время, ни разу не наблюдалось случая, чтобы кто-то из загорающих помешал другому занять привычное место.

Этот феномен так заинтересовал профессора, что тот решил провести более детальное исследование. В нём он ввёл координатную систему и для каждого из n отдыхающих отметил места, в которых они располагают свои коврики. Система координат введена таким образом, что ось X параллельна сторонам a , а ось Y — сторонам b . Изначально профессор собирался вычислить для каждой пары отдыхающих площадь пересечения территории, обычно занимаемой их ковриками. Однако впоследствии выяснилось, что для дальнейших исследований ему необходима только средняя величина соответствующих значений. Иначе говоря, профессора интересует математическое ожидание площади пересечения ковриков у двух случайно выбранных различных отдыхающих.

По представленным профессором данным вычислите соответствующую величину.

Input

Первая строка входного файла содержит три целых числа n , a и b ($2 \leq n \leq 200\,000$, $1 \leq a, b \leq 1\,000\,000$) — количество отдыхающих жителей столицы и размеры пляжного коврика. Далее заданы координаты ковриков: i -я из последующих n строк содержит два целых числа x_i и y_i ($0 \leq x_i, y_i \leq 1\,000\,000$) — координаты левого верхнего угла для коврика i -го отдыхающего.

Output

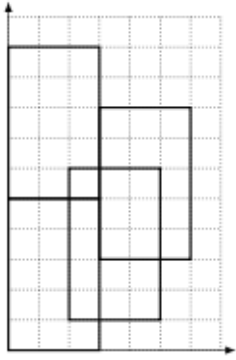
Выведите одно число — среднее значение площади пересечения двух ковриков с абсолютной точностью не хуже 10^{-2} .

Examples

standard input	standard output
4 3 5 0 0 2 1 3 3 0 5	1.833333333

Note

Для приведённого в условии задачи примера точный ответ равен $\frac{4+0+0+1+6+0}{6} = 1\frac{5}{6}$.



Problem L. Wipe! (Division 2 Only!)

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

Джон хочет заняться удалением секретных данных с жёстких дисков. Он написал программу, которая запрашивает имя файла и количество раз N , которое файл должен быть перезаписан. По замыслу Джона, этот параметр должен обозначать степень секретности. Джон обрабатывает файл в N проходов бит за битом, заменяя 0 на 1 и 1 на 0.

Джон хочет, чтобы Вы оттестировали его программу, проверив, возможно ли, что в результате удаления первого из заданных файлов получится второй. Файлы представлены как последовательность битов.

Input

Первая строка входа содержит целое число $1 \leq N \leq 20$ — степень секретности. Далее идут две непустые строки одинаковой длины, состоящие из 0 и 1 — последовательности битов в исходном и полученном файле. Длина каждой строки не превосходит 1000.

Output

Если второй файл может быть получен из первого программой Джона, выведите “Deletion succeeded”, если нет — “Deletion failed”.

Examples

standard input	standard output
1 10001110101000001111010100001110 01110001010111110000101011110001	Deletion succeeded
20 0001100011001010 0001000011000100	Deletion failed

Problem M. Timebomb (Division 2 Only!)

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

При обезвреживании бомбы с таймером, обнаруженной в пивном баре, группа спецназа Байтландии встретила следующий вариант защиты механизма от вмешательства: выключение бомбы возможно, только если на экране задано число, делящееся на 6. Если же на экране задано число, не делящееся на 6, или вообще не число, то бомба при попытке выключения взорвётся.

По заданному состоянию дисплея таймера вычислите, насколько безопасно выключать бомбу в данный момент.

Input

Вход задаёт дисплей таймера. Код на дисплее содержит от 2 до 8 знакомест. Каждая цифра представлена 5 строками и 3 столбцами из символов (допустимы пробел и звёздочка '*'). Гарантируется, что других символов во вводе нет. Знакоместа разделены одним столбцом, составленным из пробелов. После последнего знакоместа идут переводы строки.

Внутри знакоместа могут быть как цифры, так и последовательность пикселей, цифрой не являющаяся. Формат цифр представлен ниже.

```
### ### ### ### ### ### ### ### ###
***  * *** ** * * *** ** ** **
* *  *  * * * *  *  * * * *
* *  * *** ** *** ** **  * ***
* *  * *      *  * * *  * * *
***  * *** **  * *** **  * ***
```

Решётки ('#') обозначают колонки, используемые для знакомест. Ведущие нули допустимы, знак «минус» отсутствует.

Output

Если бомбу выключать безопасно, выведите "BEER!". Иначе выведите "BOOM!".

Examples

standard input	standard output
<pre> *** * * * *** ** * * * * * * * * * * * * *** ** * ** * * * * * * * * *** * * *** ** * </pre>	BEER!!
<pre> * * *** ** * * * ** * * * * * *** ** * * * * * * * * * *** ** * </pre>	BOOM!!
<pre> *** ** * *** * * * * * * * *** * * * *** * * * * * * *** ** * *** * </pre>	BOOM!!
<pre> *** *** ** * * * * * * * *** * *** ** * * * * * *** * *** * </pre>	BEER!!

Problem N. Planting Trees (Division 2 Only!)

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

Фермер Джон недавно купил n саженцев и собирается высадить их на своём участке. Фермер тратит один день на посадку одного саженца, и для каждого дерева Джон знает, сколько времени саженец будет расти до максимального размера.

Джон собирается пригласить своих друзей-фермеров посмотреть сад, когда все деревья вырастут. Определите первый день, в который Джон может их пригласить.

Input

В первой строке входного файла задано одно целое число N ($1 \leq N \leq 100\,000$) — количество саженцев. Во второй заданы N целых чисел t_i ($1 \leq t_i \leq 1\,000\,000$), где t_i задаёт количество дней, которое растёт i -й саженец.

Output

Выведите одно число — день, в который Джон может позвать соседей. Дни занумерованы с 1, начиная с текущего дня.

Examples

standard input	standard output
4 2 3 4 3	7
6 39 38 9 35 39 20	42

Problem O. Number Trick (Division 2 Only!)

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

Якуб собирается показывать некоторые математические шутки. Например, он хочет показать, что для того, чтобы умножить число на $X = 2.6$, достаточно просто перенести первую цифру в конец числа: $135 \times 2.6 = 351$, а также $270270 \times 2.6 = 702702$.

Якуб хочет показать, что метод работает для каждого X . Для этого он спрашивает у зрителей значение X , после чего показывает пример умножения, для которых этот способ работает.

По заданному X выведите все целые положительные числа, состоящие из 8 и менее цифр, для которых этот метод работает (умножение на X эквивалентно перестановке первой цифры в конец числа).

Input

В первой строке задано число X ($1 \leq X < 1000$), количество знаков после запятой не превышает 4.

Output

Выведите в возрастающем порядке все числа, меньшие, чем 10^8 , для которых шутка Якуба проходит. Если список пуст, выведите одну фразу "No solution".

Examples

standard input	standard output
2.6	135 270 135135 270270
3.1416	No solution