

Problem A. Максимальная разность

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Вася выбирает два натуральных числа, стандартная запись каждого из которых в системе счисления с основанием b состоит из N цифр, а сумма цифр в этих записях одинакова. Вася хочет сделать выбор таким образом, чтобы разность между числами была максимально возможной. Помогите Васе найти эту разность.

Input

Единственная строка содержит два целых числа b и N ($2 \leq b \leq 36$, $1 \leq N \leq 10^5$).

Output

Выведите единственное целое число — максимальную разность между N -значными числами в системе счисления с основанием b с одинаковой суммой цифр. Результат должен быть представлен в стандартной записи в системе с основанием b . Для цифр, больших 9, используйте большие латинские буквы: 'A', 'B', 'C', ..., 'Z'.

Examples

standard input	standard output
10 2	72
16 3	E01

Problem B. Построение многоугольника

Input file: *standard input*
Output file: *standard output*
Time limit: 5 секунд
Memory limit: 64 мегабайта

Дана последовательность из N целых чисел a_1, a_2, \dots, a_N . Требуется построить выпуклый многоугольник с указанными длинами сторон, никакие три вершины которого не лежат на одной прямой.

Input

Первая строка содержит целое число $3 \leq N \leq 10^5$. Во второй строке записаны целые числа a_1, a_2, \dots, a_N , $-10^5 \leq a_i \leq 10^5$.

Output

Выведите N строк. В каждой строке должны находиться декартовы координаты соответствующей вершины треугольника. Расстояние между вершинами с номерами i и $i + 1$ должно быть равно a_i , между вершинами с номерами N и 1 – значению a_N . Все равенства должны выполняться с точностью не менее 10^{-5} . Все координаты не должны превышать $2 \cdot 10^9$ по абсолютной величине. Если построить многоугольник с требуемыми свойствами невозможно, выведите одну строку “Impossible”.

Examples

standard input	standard output
3 3 4 5	0.0000000 3.0000000 0.0000000 0.0000000 4.0000000 0.0000000
3 1 2 3	Impossible
5 2 5 13 10 10	0.0000000 0.0000000 2.0000000 0.0000000 5.0000000 4.0000000 0.0000000 16.0000000 -6.0000000 8.0000000

Problem C. Красивые узоры

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайт

Компания BrokenTiles планирует заняться выкладыванием в некотором дворе узора из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Двор имеет форму прямоугольника $N \times M$ метров. Однако оказалось, что K плиток уже расположены в некоторых клетках двора. Требуется определить, сколько вариантов узора может быть выложено на этом дворе, при условии, что узор должен быть красивым.

Узор считается красивым, если в любом квадрате 2×2 есть либо три черных плитки и одна белая, либо, наоборот, одна черная и три белых плитки.

Input

В первой строке даны три целых числа N , M и K ($1 \leq N, M \leq 10^5$, $0 \leq K \leq 10^5$). В каждой из последующих K строк записаны по три целых числа x , y , c ($1 \leq x \leq N$, $1 \leq y \leq M$), обозначающих, что в клетке с координатами (x, y) находится плитка. Если она черная, то $c = 0$, а если белая, то $c = 1$. Никакие две пары (x, y) не совпадают.

Output

Выведите одно число — остаток от деления на $10^9 + 7$ количества различных красивых узоров, которые могут быть выложены на данном дворе. Узоры считаются различными, если есть хотя бы одна клетка, в которой в одном узоре лежит белая плитка, а в другом — черная.

Examples

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

Problem D. Строка без повторений

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Рассмотрим строку символов. Будем говорить, что в строке $s_1s_2\dots s_n$ есть повторение, если в ней есть две совпадающие подстроки, следующие непосредственно одна за другой — то есть для некоторых i и k ($i, k > 0, i + 2k - 1 \leq n$) выполняется $s_i = s_{i+k}, s_{i+1} = s_{i+k+1}, \dots, s_{i+k-1} = s_{i+2k-1}$.

Найдите строку длины n без повторений с минимальным количеством использованных букв.

Input

В единственной строке задается одно целое число n ($1 \leq n \leq 4 \cdot 10^6$) — длина искомой строки.

Output

В первой строке выведите минимальное количество различных символов, которые нужно использовать для построения строки без повторений, а во второй строке — искомую строку. Разрешается использовать лишь строчные латинские буквы. Гарантируется, что при всех входных данных 26 символов будет достаточно для построения строки без повторений (возможно, что не оптимальной).

Examples

standard input	standard output
5	3 abaca

Problem E. Сбор бобов

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Есть бесконечная в одну сторону полоса, состоящая из клеток, пронумерованных числами $0, 1, 2, \dots$. В некоторых клетках лежит некоторое количество бобов. Каждый ход разрешается выполнять следующее действие. Если в клетке с номером i ($i > 0$) есть не менее i горошин, то i из них забираются из этой клетки и раскладываются по одной в клетки с номерами $i-1, i-2, \dots, 0$. Если есть несколько таких клеток, то можно делать ход из любой из них. Ваша задача — собрать в клетке с номером 0 все имеющиеся на полосе бобы.

Input

В первой строке задано одно целое число N ($0 \leq N \leq 2 \cdot 10^5$). В каждой из последующих N строк задаются по два целых числа i_k и a_k ($0 \leq i_k \leq 10^9$, $1 \leq a_k \leq 10^{18}$), означающих, что в клетке с номером i_k находится в начальном состоянии a_k горошин. Гарантируется, что все i_k различны.

Output

Выведите “Yes”, если существует такая последовательность действий, которая приведет к тому, что все горошины окажутся в клетке 0. В противном случае нужно вывести “No”.

Examples

standard input	standard output
2 1 1 2 2	Yes
3 1 1 2 2 3 3	No
4 0 3 1 3 2 3 3 3	Yes

Problem F. Обратный сбор бобов

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Дана бесконечная в одну сторону полоса, состоящая из клеток, пронумерованных числами $0, 1, 2, \dots$. У вас есть K горошин, которые вы должны разместить в клетках этой полосы. Необходимо сделать это таким образом, чтобы существовала последовательность действий, которая приведет к тому, что все K бобов окажутся в клетке с номером 0. Напомним, что разрешенным является следующее действие. Если в клетке с номером i ($i > 0$) есть не менее i горошин, то i из них забираются из этой клетки и раскладываются по одной в клетки с номерами $i-1, i-2, \dots, 0$. Если есть несколько таких клеток, то можно делать ход из любой из них.

Input

В единственной строке записано одно целое число K ($0 \leq K \leq 10^{11}$).

Output

Выведите N строк, где N – количество ячеек, в которых будет ненулевое количество горошин. В каждой строке должны быть указаны два целых числа i_k и a_k , означающих, что в клетке с номером i_k находится в начальном состоянии a_k горошин. Строки должны выводиться в порядке увеличения i_k .

Если существует несколько вариантов расположения, выберите антилексикографически минимальный. Напомним, что набор $(x_0, x_1, \dots, x_i, \dots)$ антилексикографически предшествует набору $(y_0, y_1, \dots, y_i, \dots)$, если существует такое i , что $x_i > y_i$ и $x_j = y_j$ для всех $j > i$.

Examples

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

Problem G. Уравнение

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 64 мегабайта

Имеется уравнение

$$a_N x^N + a_{N-1} x^{N-1} + \dots + a_1 x + a_0 = y p^2,$$

в котором a_i , N и p — известные целые числа, а x и y — неизвестные целочисленные переменные, причем $x \geq 0$. Требуется проверить, существует ли у данного уравнения решение в целых числах.

Input

В первой строке записаны два целых числа p и N ($0 \leq N \leq 20$, $2 \leq p \leq 10^6$, p — простое число). Вторая строка содержит $N + 1$ чисел a_N, \dots, a_0 ($|a_i| \leq 10^9$).

Output

Если уравнение не имеет решений в целых числах, выведите число -1 . В противном случае выведите значение x из пары (x, y) , удовлетворяющей уравнению. Если таких решений несколько, выберите такое, в котором x принимает наименьшее неотрицательное значение.

Examples

standard input	standard output
3 2 1 2 3	3
2 2 1 1 1	-1

Problem H. Соревнование

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

На соревновании по карате участвуют две команды A и B . В каждой из них есть по N каратистов с известным уровнем мастерства. Соревнование состоит из N раундов-поединков, в каждом из которых участвует по одному каратисту от каждой команды. Согласно правилам, каждый каратист может поучаствовать только в одном поединке. Каждый раунд заканчивается победой одного из участников (того, у которого выше уровень мастерства), либо ничьей (если уровни мастерства участников одинаковы). Если зафиксирована ничья в поединке, обе команды за раунд получают по 1 очку, в противном случае победитель приносит своей команде 2 очка, проигравший – 0. В начале соревнования тренеры обеих команд дают организаторам списки, где указывают, в каком порядке будут выходить на ринг их подопечные. Однако тренеру команды B стало известно, в каком порядке будут выходить участники из команды A . Обладая этой информацией, он хочет составить свой список таким образом, чтобы набрать как можно больше очков, и просит вас помочь ему.

Input

В первой строке задается одно целое число N ($1 \leq N \leq 2 \cdot 10^5$). Во второй строке задаются N чисел a_i , каждое из которых определяет уровень мастерства соответствующего каратиста команды A . Значения задаются в том порядке, в котором будут выходить на ринг участники этой команды. В третьей строке задаются N чисел b_i , определяющих уровни мастерства каратистов команды B ($1 \leq a_i, b_i \leq 10^9$).

Output

Выведите одно число — максимальное количество очков, которое может набрать команда B за счет выбора тренером этой команды оптимального порядка.

Examples

standard input	standard output
3 3 4 5 1 2 6	2
4 4 5 6 2 1 7 3 8	6

Note

Во втором примере в первом раунде тренер B выпустит каратиста с уровнем 1 (который наберет 0 очков в поединке с каратистом уровня 4), а в последующих – каратистов с уровнями 7, 8 и 3 соответственно, которые выиграют свои поединки против 5, 6 и 2 и принесут по два очка.

Problem I. Вписанная окружность

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Задан выпуклый многоугольник. Будем говорить, что некоторая окружность является вписанной в данный многоугольник, если все ее точки содержатся внутри многоугольника или на его границе. Требуется найти вписанную окружность с наибольшим радиусом.

Input

В первой строке задается одно целое число N . В каждой из последующих N ($3 \leq N \leq 50000$) строк содержатся по два вещественных числа x_i, y_i ($-10^7 \leq x_i, y_i \leq 10^7$), имеющих не более 6 знаков после десятичной точки и определяющие координаты соответствующей вершины многоугольника. Вершины многоугольника задаются в порядке обхода. Никакие три точки не лежат на одной прямой.

Output

Выведите одно число – радиус максимальной вписанной окружности с точностью не менее 10^{-5} .

Examples

standard input	standard output
3 2.0 0.0 0.0 0.0 1.0 1.0	0.414214
4 -1 0 0 1 2 -1 1 -2	0.707107

Problem J. Разделяющая прямая

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

Задано множество, состоящее из N точек на плоскости. Прямая называется разделяющей по отношению к данному множеству, если найдутся две точки этого множества, лежащие в разных полуплоскостях относительно этой прямой (но не на самой прямой). Ваша задача — определить для каждой из заданных прямых, является ли она разделяющей или нет.

Input

В первой строке задается целое число N ($0 \leq N \leq 10^5$). В каждой из последующих N строк содержатся по два целых числа x_i, y_i , определяющие координаты соответствующей точки множества. В $(N + 2)$ -ой строке задается число M ($0 \leq M \leq 10^5$), а в последующих M строках — по 4 целых числа X_1, Y_1, X_2, Y_2 , где (X_1, Y_1) и (X_2, Y_2) — две различные точки на соответствующей прямой.

Все координаты — целые числа, не превосходящие 10^9 по абсолютной величине.

Output

Выведите M строк, каждая из которых определяет результат для соответствующей прямой. Если прямая является разделяющей, строка должна содержать два числа — номера (начиная с единицы) каких-либо точек множества, лежащих в разных полуплоскостях относительно прямой. Если же прямая не является разделяющей, строка должна содержать одно число 0.

Examples

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

Problem K. Стрингангуляция (высшая лига)

Input file: *standard input*
Output file: *standard output*
Time limit: 1.5 секунды
Memory limit: 64 мегабайта

Пусть задана некоторая строка $s_1s_2\dots s_l$. Предполагаем, что она циклическая, то есть за s_l следует символ s_1 . Будем называть стрингангуляцией этой строки ее разбиение на три последовательные подстроки (уже не циклические, обозначим их соответственно a , b и c), для каждой из которых выполнено неравенство треугольника, то есть

$$a + b > c, b + c > a, c + a > b,$$

где знаком “+” понимается, как обычно для строк, их конкатенация. Строки сравниваются лексикографически.

Ваша задача — найти для заданной строки количество различных ее стрингангуляций. Стрингангуляции, отличающиеся лишь обозначениями последовательных подстрок, не считаются различными.

Input

Единственная строка определяет строку s , состоящую из строчных латинских букв. Длина строки s находится в диапазоне от 3 до 2014.

Output

Выведите одно число — количество различных стрингангуляций строки s .

Examples

standard input	standard output
aaa	1
cbccbcacb	2
stringangulation	0

Problem L. Построение куба

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 64 мегабайта

В трехмерном пространстве задана плоскость уравнением $Ax + By + Cz + D = 0$. Требуется построить куб, вершины которого удалены от этой плоскости на заданные расстояния d_1, d_2, \dots, d_8 .

Input

В первой строке задаются коэффициенты уравнения плоскости A, B, C и D , а во второй строке расстояния d_1, d_2, \dots, d_8 .

Все значения — вещественные, не превышают 10^4 по абсолютной величине и имеют не более двух знаков в дробной части. Хотя бы одно из чисел A, B и C отлично от нуля, все значения d_i — неотрицательны.

Output

Выведите 8 строк, каждая из которых будет содержать по три числа — координаты x, y и z соответствующей вершины куба. Первая выведенная точка должна находиться на расстоянии d_1 от плоскости (с точностью до 10^{-5}), вторая — на расстоянии d_2 и т.д.

Examples

standard input	standard output
1.0 1.0 1.0 1.0 1.0 3.0 2.0 4.0 2.0 3.0 3.0 2.0	0.244017 0.244017 0.244017 1.976068 1.976068 0.244017 1.976068 0.244017 0.244017 1.976068 1.976068 1.976068 0.244017 1.976068 0.244017 1.976068 0.244017 1.976068 0.244017 1.976068 1.976068 0.244017 0.244017 1.976068
-1 2 -3 6 0 0 0 0 0 0 0 0	Impossible