

Problem B. Dominating Set

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 64 mebibytes

У Бобо есть двудольный граф $G = (V, E)$ с n вершинами и m рёбрами. Он хочет выбрать подмножество D вершин такое, что для каждой вершины v , не входящей в D , как минимум одна из соседних с ней вершин находится в D . Найдите количество способом, которым Бобо сможет это сделать.

Напоминаем, что

1. Граф G является двудольным тогда и только тогда, когда он не содержит циклов нечётной длины.
2. Вершина u является соседней с вершиной v тогда и только тогда, когда u и v соединены ребром.

Input

Первая строка входа содержит два целых числа n и m ($1 \leq n \leq 30$, $0 \leq m \leq 225$).

i -я из последующих m строк содержит два целых числа a_i и b_i и обозначает, что вершины a_i и b_i соединены ребром ($1 \leq a_i, b_i \leq n$).

Гарантируется, что граф не содержит петель и кратных рёбер.

Output

Выведите одно целое число: количество различных подмножеств, удовлетворяющих условию задачи.

Examples

standard input	standard output
4 4 1 2 2 3 3 4 4 1	11
4 0	1

Problem F. Similar Subsequence

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 64 mebibytes

Две последовательности $\{a_1, a_2, \dots, a_n\}$ и $\{b_1, b_2, \dots, b_m\}$ называются *похожими* тогда и только тогда, когда $(a_i - a_j) \cdot (b_i - b_j) > 0$ для всех $1 \leq i, j \leq n$.

У Бобо есть две последовательности $A = \{a_1, a_2, \dots, a_n\}$ и $B = \{b_1, b_2, \dots, b_m\}$. Он хочет проверить, содержит ли B подпоследовательность, похожую на A .

Input

Первая строка входа содержит два целых числа n и m ($1 \leq n, m \leq 500$).

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$).

Третья строка содержит m целых чисел b_1, b_2, \dots, b_m ($1 \leq b_i \leq m$).

Гарантируется, что A является перестановкой $\{1, 2, \dots, n\}$ и **не содержит** подпоследовательностей, похожих на $\{2, 1, 3\}$ или же на $\{2, 3, 1\}$.

Output

Выведите “Yes”, если B содержит подпоследовательность, похожую на A , или “No” в противном случае.

Examples

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

Problem G. Random Arithmetics

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 64 mebibytes

Бобо играет с набором из n целых чисел. Изначально набор состоит из чисел a_1, a_2, \dots, a_n .

Бобо многократно применяет следующую операцию: выбирает пару целых чисел x и y , имеющуюся в наборе, и заменяет их числом $x + y$ или $x \cdot y$.

Все пары равновероятны, арифметическое действие также выбирается с вероятностью $1/2$. Например, после первой операции существует $n \cdot (n - 1)$ возможных равновероятных исходов.

После того, как Бобо повторил вышеописанную операцию $(n - 1)$ раз, в наборе осталось одно число. Требуется найти математическое ожидание этого числа.

Input

Первая строка входа содержит целое число n ($2 \leq n \leq 2000$).

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$).

Output

Если ожидаемая величина равна $\frac{P}{Q}$, выведите $P \cdot Q^{-1} \bmod (10^9 + 7)$, где Q^{-1} — число, для которого $Q \cdot Q^{-1} \equiv 1 \pmod{(10^9 + 7)}$.

Examples

standard input	standard output
2 1 1	500000005
3 1 2 3	250000008

Problem J. Welcome to ICPCCamp 2016!

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 64 mebibytes

Welcome to ICPCCamp 2016!

Бобо пытается решить одну из самых простых задач в конкурсе.

Даны 6666 целых чисел $a_1, a_2, \dots, a_{6666}$ между 1 и 2016.

Требуется найти подмножество из 2016 целых чисел, сумма которых делится на 2016.

Бобо не только решил задачу, но и заметил, что в следующий раз такая задача будет решается только в 2025 году.

А сможете ли Вы хотя бы решить эту задачу?

Input

Входной файл состоит из 6666 строк, каждая из которых содержит одно целое число a_i ($1 \leq a_i \leq 2016$).

Output

Выведите 2016 попарно различных чисел $b_1, b_2, \dots, b_{2016}$ таких, что набор из $a_{b_1}, a_{b_2}, \dots, a_{b_{2016}}$ является ответом к задаче.

Если ответов несколько, выведите любой из них.

Example

standard input	standard output
1	1
1	2
1	3
...(6660 lines omitted)	...(2010 lines omitted)
1	2014
1	2015
1	2016

Problem K. Ant's way

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 64 mebibytes

На плоскости задан простой многоугольник (многоугольник называется простым, если он имеет ненулевую площадь, никакие две стороны не имеют более одной общей точки и никакие две стороны, кроме соседних, не имеют общих точек). Муравей сидит в точке (x_0, y_0) и начинает движение в направлении вектора (v_x, v_y) .

Вычислите длину части пути муравья, лежащей строго внутри многоугольника.

Input

Первая строка входа содержит целое число N — количество вершин многоугольника ($1 \leq N \leq 100$). Каждая из последующих N строк содержит координаты очередной вершины многоугольника x_i и y_i ; координаты перечислены в порядке обхода. Последняя строка содержит координаты начальной точки муравья (x_0, y_0) и вектора (v_x, v_y) , вдоль которого движется муравей.

Все координаты — вещественные числа, заданные не более, чем с тремя знаками после десятичной точки и не превосходящие 1000 по абсолютной величине.

Output

Выведите одно число — длину части пути муравья, лежащей строго внутри многоугольника, с абсолютной или относительной погрешностью не хуже 10^{-5}

Examples

standard input	standard output
4 0 0 0 1 1 1 1 0 -1 -1 1 1	1.4142135624 говёнка
4 0 0 0 1 1 1 1 0 -1 0 1 0	0.00000
5 0 0 1 10 2 1 3 10 4 0 -1 1 1 0	3.8000000000

Problem L. Cut The Rectangle

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 64 mebibytes

Два треугольника заданы длинами сторон. Определите, можно ли получить эти треугольники, разрезав прямоугольник одним отрезком на две части и затем, возможно, повернув получившиеся части каким-либо образом или отразив их.

Input

Первая строка входа содержит три целых числа — стороны первого треугольника. Вторая строка входа содержит три целых числа — стороны второго треугольника.

Гарантируется, что заданные тройки чисел могут быть сторонами треугольника ненулевой площади. Длина каждой стороны треугольника не менее 1 и не более 100.

Output

Если существует прямоугольник, который может быть разрезан так, чтобы получились два заданных треугольника, выведите 1. В противном случае выведите 0.

Examples

standard input	standard output
6 8 10 8 6 10	1
7 4 5 4 5 7	0

Problem M. Diversity

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 64 mebibytes

Определим *разнообразность* строки как количество попарно различных букв в строке. Например, строка “acm” имеет разнообразность 3, равно как и строка “icpc”.

Бобо нравятся строки, разнообразность которых равна 1 или 2. Ему подарили некоторую строку; Бобо хочет превратить её в строку, которая ему нравится. За одно действие он может удалить одну букву в любом месте строки. За какое минимальное количество действий Бобо сумеет превратить данную строку в строку с разнообразностью 2 или менее?

Input

На вход подаётся непустая строка, состоящая не более, чем из 100 строчных латинских букв.

Output

Выведите одно целое число — наименьшее количество действий, которые потребуются Бобо, чтобы превратить данную строку в строку, которая ему нравится.

Examples

standard input	standard output
bobo	0
china	3
acmicpc	3

Problem N. Mechanics

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 64 mebibytes

На плоскости размещён набор из n зубчатых колёс. Центр каждого зубчатого колеса находится в точке с целыми координатами, радиус также является целым. Требуется вычислить, что будет происходить с последним перечисленным колесом при попытке вращения первого. Есть следующие варианты:

- Первое колесо заблокировано и не может двигаться, так как какое-то колесо при его вращении должно будет вращаться в две противоположные стороны одновременно.
- Первое колесо может двигаться, однако оно не соединено с последним колесом, которое тем самым остаётся неподвижным.
- Первое колесо приводит последнее в движение с определённым передаточным отношением.

В случае, если первое колесо заблокировано, выведите эту информацию вне зависимости от того, соединено ли первое колесо с последним.

Input

Первая строка входа содержит целое число n ($1 \leq n \leq 1,000$) — количество зубчатых колёс. Далее следуют n строк, каждая из которых задаёт одно зубчатое колесо и содержит три целых числа x , y ($-10^4 \leq x, y \leq 10^4$) и r ($1 \leq r \leq 10^4$) — координаты оси колеса и его радиус. Первое колесо (которое вращают) перечислено первым, колесо, информацию о движении которого надо вывести, перечислено n -м.

Считается, что два любых зубчатых колеса, которые касаются, зацеплены между собой. Гарантируется, что никакие два зубчатых колеса не пересекаются по фигуре ненулевой площади.

Output

Выведите строку со следующим содержанием:

- -1 , если первое колесо заблокировано.
- 0 , если первое колесо может вращаться, но при этом последнее колесо остаётся неподвижным.
- $a\ b$, если при вращении первого колеса последнее тоже вращается; a и b — два целых числа, разделённых пробелом, и $a : b$ — отношение частоты вращения первого колеса к последнему, причём a и b взаимно просты, a всегда положительно, а b положительно, если последнее колесо вращается в ту же сторону, что и первое, и отрицательно в противном случае.

Example

standard input	standard output
2 0 0 10 0 30 20	2 -1
2 0 0 1 0 3 1	0
3 0 0 11 0 33 22 44 0 33	-1

Problem O. Pairs

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 64 mebibytes

Карточная игра «Pairs» играется специальными картами. Каждой карте соответствуют два целых числа: одно задаёт масть, второе задаёт цену. Карты в колоде могут повторяться.

Игрок получает от сдающего некоторый набор карт. После чего он для каждой масти, карты которой присутствуют в руке, должен сбросить ровно две карты этой масти. Если в какой-то масти ровно одна карта — игрок проиграл и получил бонус 0.

Иначе бонус определяется как сумма значений карт, которые остались у игрока.

По заданному набору карт определите, какие карты нужно сбросить, чтобы максимизировать бонус.

Input

Первая строка входа содержит одно целое число N ($1 \leq N \leq 299999$) — количество карт, которое раздал сдающий. i -я из последующих N строк задаёт i -ю слева карту в руке и содержит два целых числа: масть s_i ($1 \leq s_i \leq 10^9$) и значение v_i ($0 \leq v_i \leq 10^9$).

Output

Если игрок с данным набором карт проиграет, выведите -1 . Иначе выведите в первой строке целое число K — количество мастей, которые есть у игрока, затем для каждой масти в новой строке выведите два числа — номера (при перечислении в соответствии со входным файлом, слева направо, в нумерации, начиная с 1) карт, которые должны быть сброшены в этой масти.

Порядок мастей при выводе можно выбирать произвольно. Если решений несколько, выведите любое.

Examples

standard input	standard output
10	3
9 2	3 4
9 3	1 2
7 1	8 9
7 2	
9 4	
9 3	
1 2	
1 0	
1 0	
7 3	

Problem P. Trip Between Corners

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 64 mebibytes

Фишка стоит на доске $n \times m$, на каждом поле которой записана цифра. С поля, на котором записана цифра k , можно ходить ровно на k полей влево, вправо, вверх или вниз. Выходить за пределы доски нельзя, доска не является «зацикленной».

За какое наименьшее количество ходов можно прийти от левого верхнего угла в правый нижний (если это вообще возможно)?

Input

Первая строка ввода содержит два целых числа n и m ($1 \leq n, m \leq 500$), задающих размеры поля. Гарантируется, что $mn > 1$.

Каждая из последующих n строк состоит из m цифр ('0' – '9'), записанных вплотную (без пробела); i -я цифра в j -й строке задаёт поле (i, j) .

Верхний левый угол доски соответствует первому символу первой из n строк, правый нижний — последнему символу n -й строки.

Output

Если прийти от левого верхнего угла доски в правый нижний возможно, выведите наименьшее число ходов, за которое это можно сделать, в противном случае выведите -1 .

Examples

standard input	standard output
3 3 111 111 111	3
2 2 33 33	-1