Задача A. Линейные уравнения

Имя входного файла:	`linear.in`
Имя выходного файла:	`linear.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Система линейных уравнений, как всем известно, есть множество уравнений

$$\begin{aligned} a_{11}x_1 + \ldots + a_{1n}x_n &= b_1 \\ &\ldots \\ a_{n1}x_1 + \ldots + a_{nn}x_n &= b_n \end{aligned}$$

Ваша задача — решить её.

Формат входных данных

В первой строке входного файла записано целое число n ($1 \leqslant n \leqslant 20$). В следующих n строках записано по $n + 1$ целых чисел: $a_{i1}, \ldots, a_{in}, b_i$. Все эти числа не превышают 100 по абсолютному значению.

Формат выходных данных

Первая строка выходного файла должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `infinity` — бесконечно много решений
- `single` — единственное решение. В этом случае вторая строка должна содержать n чисел $x_1, \ldots, x_n$, разделенных пробелами. Решение должно быть выведено с точностью не менее трех знаков после десятичной точки.

Примеры

linear.in	linear.out
2 1 1 1 2 2 2	infinity
2 1 2 0 1 2 1	impossible

Задача B. Обращение матрицы.

Имя входного файла:	`inverse.in`
Имя выходного файла:	`inverse.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дана квадратная матрица. Найдите обратную к ней.

Формат входных данных

На первой строке входного файла находится одно число N — размер матрицы ($1 \leqslant N \leqslant 100$). Далее следуют N строк по N вещественных чисел в каждой — матрица.

Формат выходных данных

Если обратной матрицы не существует, то выведите в выходной файл одну строку `NO`. Иначе в первой строке выходного файла выведите одно слово `YES`, а далее выведите N строк по N вещественных чисел в каждой — обратную матрицу. Ответ будет считаться правильным, если абсолютная или относительная погрешность элементов произведения будет не больше, чем 10^{-6}

Примеры

inverse.in	inverse.out
2 0 1.0 1 0	YES 0.00000000000000000000 1.00000000000000000000 1.00000000000000000000 0.00000000000000000000
2 0 0 1 0	NO

Задача C. Добавление векторов

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В изначально пустое множество один за другим добавляются m битовых векторов размера n . После добавления каждого скажите, можно ли его представить как xor векторов, добавленных до него.

Формат входных данных

В первой строке записаны числа n и m ($1 \leqslant n \leqslant 50$, $1 \leqslant m \leqslant 10000$). В следующих m строках записаны вектора.

Формат выходных данных

Для каждого вектора выведите `1`, если его можно представить как xor предыдущих и `0` если нельзя.

Примеры

стандартный ввод	стандартный вывод
2 2 10 11	0 0
3 4 100 111 011 010	0 0 1 0
3 4 000 111 111 111	1 0 1 1

Задача D. Максимизировать сумму XOR

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Будем обозначать как $\oplus$ операцию *побитового «исключающего или»* для целых чисел. В языках программирования C++ и Java она обозначается символом «`^`», в паскале и Python — ключевым словом «`xor`». Например, $9 \oplus 3 = 1001_2 \oplus 11_2 = 1010_2 = 10$.

Даны два массива A и B длины n. Обозначим как $X(A)$ для массива A результат вычисления побитового «исключающего или» от всех элеметов массива: $X(A) = A_1 \oplus A_2 \oplus \ldots \oplus A_n$. Аналогично, введем обозначение $X(B) = B_1 \oplus B_2 \oplus \ldots \oplus B_n$.

Для каждого i от 1 до n разрешается поменять местами элементы A_i и B_i. Необходимо определить, какие из этих обменов надо сделать, чтобы максимизировать сумму $X(A) + X(B)$.

Формат входных данных

В первой строке входных данных находится число n — количество элементов ($1 \leqslant n \leqslant 10^5$). В следующей строке находится n элементов массива A ($0 \leqslant A_i \leqslant 10^{18}$). В следующей строке в таком же формате дан массив B.

Формат выходных данных

В первой строке выведите максимальную возможную сумму и число k — количество необходимых обменов. В следующей строке выведите k различных чисел от 1 до n — индексы элементов, которые надо поменять.

Пример

стандартный ввод	стандартный вывод
2 1 1 2 2	6 1 1

Замечание

В примере после обмена массивы равны $A = [2, 1]$ и $B = [1, 2]$, соответственно.
$X(A) = 2 \oplus 1 = 10_2 \oplus 1_2 = 11_2 = 3$, $X(B) = 3$, $X(A) + X(B) = 6$.

Задача E. Сумма степеней

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вы, конечно, знаете формулу суммы первых n натуральных чисел:

$$1 + 2 + \ldots + n = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$$

Наверное, вы знаете формулу суммы первых n квадратов натуральных чисел:

$$1^2 + 2^2 + \ldots + n^2 = \frac{n(n+1)(2n+1)}{6} = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n$$

Обобщите на сумму первых k-х степеней.

Формат входных данных

Одно натуральное число k ($1 \leqslant k \leqslant 20$) — степень слагаемых, которые суммируются.

Формат выходных данных

Выведите $k+1$ натуральных чисел: коэффициенты при степенях $n^{k+1}, n^k, \ldots, n^1$ в искомом многочлене — формуле для суммы $1^k + 2^k + \ldots + n^k$.

Легко доказать, что коэффициент при n^0 будет всегда 0, поэтому его выводить не надо. (Если этот коэффициент не 0, то при $n = 0$ получится ненулевая сумма.)

Допускается относительная или абсолютная погрешность не более 10^{-4}.

Примеры

стандартный ввод
1
стандартный вывод
0.5 0.5

стандартный ввод
2
стандартный вывод
0.3333333333333333 0.5 0.16666666666666669

стандартный ввод
4
стандартный вывод
0.2 0.5 0.3333333333333333 -0.0 -0.03333333333333334

Задача F. Светофорчики

Имя входного файла:	`traflights.in`
Имя выходного файла:	`traflights.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Лагерь «Берендеевы дороги», что недалеко от г. Сусанино Костромской области,— это лагерь для юных инспекторов дорожного движения. Дорожки в лагере имеют следующую структуру: есть N дорожек, идущих параллельно друг другу со стороны столовой к бассейну, и M дорожек, идущих параллельно друг другу и перпендикулярно предыдущей группе дорожек параллельно стене столовой. Соответственно, в лагере есть NM перекрестков.

На каждом перекрёстке находится светофор. Каждый светофор имеет три фазы:

1. сначала горит зелёный свет по направлению столовая — бассейн и красный в другую;
2. потом горит красный свет по направлению столовая — бассейн, и зелёный в другую;
3. после этого в обе стороны загорается красный, зато загорается зелёный для пешеходов в обе стороны.

После этого цикл повторяется.

Ночью, когда все юные инспектора спят, более старшие и опытные играют со светофорами. Они отключают автоматическую смену фаз светофоров и начинают переключать их вручную. На каждом светофоре есть кнопка; при нажатии на эту кнопку этот светофор, а также все, расположенные с ним на одной дорожке, (всего $N + M - 1$ светофор) меняют свою фазу на следующую (т. е. те светофоры, которые были в фазе 1, переходят в фазу 2; те, что были в фазе 2, переходят в фазу 3; а те, что были в фазе 3, переходят в фазу 1).

Наконец все собираются спать. Но, чтобы директор лагеря, полковник милиции, мог дойти из столовой до корпуса, необходимо, чтобы все светофоры горели зелёным светом для пешеходов.

Напишите программу, которая, зная, в каком состоянии остались светофоры после игры преподавателей, найдёт какую-нибудь последовательность нажатий. Обратите внимание, что милиционеры — люди терпеливые, поэтому минимизировать число нажатий не требуется.

Формат входных данных

Первая строка входного файла содержит два числа N и M — количества дорожек в лагере ($1 \leqslant N \leqslant 14$, $1 \leqslant M \leqslant 14$). Далее следуют N строк по M чисел от 1 до 3, обозначающих фазы светофоров после игры преподавателей.

Гарантируется, что N и M таковы, что решение существует для любой комбинации фаз светофоров на поле данного размера.

Формат выходных данных

В первую строку выходного файла выведите k — количество нажатий на кнопки в вашем решении, а далее выведите k строк по два числа в каждой — координаты светофора (номер строки и столбца во входном файле, считая с 1), кнопку на котором надо нажать. Число нажатий не должно превосходить $3NM$.

Примеры

traflights.in	traflights.out
3 2 2 1 3 1 2 1	2 1 2 3 2

Задача G. Квадратное уравнение

Имя входного файла:	`quadratic.in`
Имя выходного файла:	`quadratic.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Детей в школе учат решать квадратные уравнения, т.е. уравнения вида

$$ax^2 + bx + c = 0,$$

где a, b и c некоторые заданные действительные числа, а x — действительное число, которое необходимо найти.

В этой задаче вам потребуется решить квадратное уравнение для многочленов с коэффициентами из нулей или единиц, и все операции производятся по модулю 2.

Даны многочлены $a(t)$, $b(t)$ и $c(t)$, найдите такой полином $x(t)$ что

$$a(t)x^2(t) + b(t)x(t) + c(t) = 0,$$

где равенство понимается как равенство многочленов. Напомним, что многочлены равны тогда и только тогда, когда равны их коэффициенты при соответствующих степенях t.

Формат входных данных

Входной файл содержит многочлены $a(t)$, $b(t)$ и $c(t)$, которые задаются их степенями, за которыми следуют коэффициенты, начиная со старшего. Нулевые многочлены в данной задаче имеют степень -1. Степени всех многочленов не превосходят 127. Между старшим коэффициентом и степенью находится **два** пробела. После многочлена степени -1 также находится один пробел.

Формат выходных данных

Если есть хотя бы одно решение уравнения, выведите любое из них в таком же формате. Старший коэффициент найденного многочлена не должен быть нулевым. Степень полинома не должна превышать 512.

В противном случае напечатайте “`no solution`”.

Примеры

quadratic.in	quadratic.out
0 1 2 1 1 0 3 1 0 0 0	1 1 0