

Задача А. 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

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

Ваша задача — решить её.

Формат входных данных

В первой строке входного файла записано целое число n ($1 \leq n \leq 20$). В следующих n строках записано по $n + 1$ целых чисел: $a_{i1}, \dots, a_{in}, b_i$. 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, \dots, x_n , разделенных пробелами. Решение должно быть выведено с точностью не менее трех знаков после десятичной точки.

Примеры

<code>linear.in</code>	<code>linear.out</code>
2 1 1 1 2 2 2	<code>infinity</code>
2 1 2 0 1 2 1	<code>impossible</code>

Задача В. 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 ($1 \leq N \leq 100$). 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

<code>inverse.in</code>	<code>inverse.out</code>
2 0 1.0 1 0	YES 0.00000000000000000000 1.00000000000000000000 1.00000000000000000000 0.00000000000000000000
2 0 0 1 0	NO

Задача С. 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:

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

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

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

Обобщите на сумму первых k -х степеней.

Формат входных данных

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

Формат выходных данных

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

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 D. 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 хог векторов, добавленных до него.

Формат входных данных

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

Формат выходных данных

Для каждого вектора выведите 1, если его можно представить как хог 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 Е. Головоломка

Имя входного файла: puzzle.in
Имя выходного файла: puzzle.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вася очень любит играть в квесты. Он любит кликать тут и там мышкой и наслаждается шутками, которых много в квестах. Но там есть еще и много разных головоломок, и Вася проводит много времени, решая их. Но однажды он столкнулся с головоломкой которую никак не мог решить. К счастью, Вася отличный программист, и он смог написать программу, которая решила головоломку и помогла ему закончить квест.

Васина головоломка представляет собой матрицу 3×3 , каждая клетка которой окрашена в черный или белый цвет. Если кликнуть на клетку, то она и ее соседи меняют свой цвет на противоположный. Цель — сделать все клетки одного цвета.

Ваша задача чуть более сложная. Пусть есть N клеток, занумерованных от 1 до N . Каждая клетка имеет множество клеток, связанных с ней. Когда игрок кликает на клетку, все клетки из множества, связанного с ней, меняют свой цвет. По данным связанным множествам и начальной раскраске выведите последовательность клеток, на которые нужно кликать, чтобы все клетки приобрели один цвет. Если есть несколько различных решений, выведите любое.

Формат входных данных

Первая строка содержит единственное целое число N ($1 \leq N \leq 200$). В $i+1$ -й строке ($1 \leq i \leq N$) находится описание множества, связанного с i -й клеткой. Каждое описание начинается с целого числа k — количество клеток в множестве, а затем идут k различных целых чисел (номера клеток). Последняя строка содержит N нулей и единиц — начальная раскраска клеток.

Формат выходных данных

Если невозможно окрасить все клетки в один цвет, выведите единственное число -1 . Иначе выведите целое число L — количество кликов, необходимых для решения головоломки, а затем L чисел — номера клеток, на которые необходимо кликать.

Примеры

puzzle.in	puzzle.out
9	9
3 1 2 4	1 2 3 4 5 6 7 8 9
4 1 2 3 5	
3 2 3 6	
4 1 4 5 7	
5 2 4 5 6 8	
4 3 5 6 9	
3 4 7 8	
4 5 7 8 9	
3 6 8 9	
1 0 1 0 1 0 1 0 1	

Задача F. 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	1 1 0
2 1 1 0	
3 1 0 0 0	