

Содержание

Обязательные задачи	2
Задача А. Линейные уравнения [0.2 sec, 256 mb]	2
Задача В. Головоломка [0.2 sec, 256 mb]	3
Задача С. Обратная матрица [0.5 sec, 256 mb]	4
Задача D. Никифор [0.5 sec, 256 mb]	5
Бонус	6
Задача Е. Разложение матрицы [2.5 sec, 256 mb]	6
Задача F. Игровой автомат [0.2 sec, 256 mb]	7
Задача G. Векторы [1 sec, 256 mb]	8
Задача H. Изоморфизм графов [1 sec, 256 mb]	9
Задача I. Шалтай-Болтай [0.3 sec, 256 mb]	10

В некоторых задачах большой ввод и вывод. Имеет смысл пользоваться супер быстрым вводом-выводом:

http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write.cpp.html

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу:

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

Обязательные задачи

Задача А. Линейные уравнения [0.2 sec, 256 mb]

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 ровно с тремя знаками после десятичной точки.

Примеры

linear.in	linear.out
2 1 1 1 2 2 2	infinity
2 1 2 0 1 2 1	impossible
2 1 2 1 2 1 0	single -0.333 0.667

Подсказка по решению

Есть у вас WA 34, это проблемы с точностью.

Задача В. Головоломка [0.2 сек, 256 mb]

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.

Пример

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 С. Обратная матрица [0.5 sec, 256 mb]

Дана матрица A размера $n \times n$, состоящая из нулей и единиц. Найти матрицу B размера $n \times n$, состоящую из нулей и единиц: $A \cdot B = E \pmod 2$, где E – единичная матрица.

Формат входных данных

На первой строке число n ($1 \leq n \leq 1000$).

Следующие n строка содержат по n нулей и единиц – описание матрицы A .

Формат выходных данных

На первой строке выведите YES, если B существует, и NO иначе. Если B существует, на следующих N строках выведите матрицу B . Если ответов несколько, выведите любой.

Примеры

invmat.in	invmat.out
2 01 11	YES 11 10
2 00 10	NO

Задача D. Никифор [0.5 сек, 256 mb]

Никифор решил подарить декану факультета линейно независимую систему векторов. Магазин продает m n -мерных векторов, $3 \leq n \leq m \leq 2000$; $n \leq 50$. Для каждого вектора известна его цена c_i , $0 < i \leq m$. Никифор хочет купить n линейно независимых векторов за минимальную сумму денег. Напишите программу, которая будет определять, какие векторы Никифор должен купить или сообщить, что это невозможно.

Формат входных данных

Первая строка входного файла содержит числа m и n через пробел. Следующие m строк содержат векторы на продажу. Все координаты являются целыми числами и не превосходят по модулю 2000. Следующие m строк содержат цены c_i . Цены — положительные целые числа, не превышающие 15000.

Формат выходных данных

Первая строка выходного файла должна содержать минимальное количество денег, которые Никифор должен заплатить или число 0, если требования Никифора не могут быть удовлетворены. Если возможно сделать покупку, то на следующих n строках выведите номера векторов, которые Никифор должен купить. Если вариантов ответа несколько, выведите минимальный в лексикографическом порядке.

Примеры

nikifor.in	nikifor.out
5 3	40
1 0 0	1
0 1 0	2
0 0 1	4
0 0 2	
0 0 3	
10	
20	
30	
10	
10	

Бонус

Задача Е. Разложение матрицы [2.5 сек, 256 mb]

Дана матрица размера $n \times t$, в которой есть хотя бы один ненулевой элемент. Необходимо разложить её в произведение двух матриц размера $n \times k$ и $k \times t$, так, чтобы k было минимально. Все операции производятся по простому модулю p .

Формат входных данных

Во первой строке входного файла заданы три числа — n , t и p ($1 \leq n \leq 300$, $1 \leq t \leq 300$, $1 \leq p \leq 10^6$, p — простое). В следующих n строках записаны по t чисел от 0 до $p - 1$ — элементы матрицы. Все числа во входном файле целые.

Формат выходных данных

В первую строку выведите одно число — k . Далее в n строках выведите по k чисел от 0 до $p - 1$ — элементы первого сомножителя. В следующих k строках выведите по t чисел от 0 до $p - 1$ — элементы второго сомножителя. Пробелы можно расставлять произвольным образом.

Пример

decomposition.in	decomposition.out
3 4 239	2
1 2 3 4	1 2
10 8 5 20	10 8
43 38 29 92	43 38
	1 0 198 160
	0 1 22 161

Задача F. Игровой автомат [0.2 sec, 256 mb]

Создатели игровых автоматов выпустили на рынок математическую игру. Игра состоит в следующем: в каждом автомате создателями задана уникальная комбинация из $M + 1$ числа в P -ичной системе счисления. Каждое число состоит из N разрядов. Первые M чисел во время игры не меняют своего значения, обозначим их a_1, a_2, \dots, a_M . Во время игры автомат несколько раз случайным образом выбирает из первых M чисел одно число и поразрядно прибавляет его к $M + 1$ -му числу (будем называть $M + 1$ -е число счетчиком) по модулю P , тем самым, изменяя его (т. е. счетчик поразрядно накапливает сумму всех сложений по модулю P). Игрок получает выигрыш, если в результате игры счетчик обнулится.

Поразрядное сложение по модулю P выполняется следующим образом: если в каком-либо разряде числа получено значение, большее $P - 1$, то оно уменьшается на P , например, при $P = 5$ и $N = 3$ результат сложения чисел 123 и 144 равен 212.

Вам прислали на инспекцию несколько таких автоматов, удостоверьтесь в том, что выигрыш принципиально возможен.

Формат входных данных

Первая строка файла содержит натуральное число — количество автоматов, присланных на инспекцию. В следующих строках описываются сами игровые автоматы. Каждый автомат описывается отдельно в следующем формате. Первая строка содержит числа P, N, M ($1 \leq N, M \leq 100, 2 \leq P \leq 255$). Следующие $M + 1$ строк содержат по N чисел в P -ичной системе счисления, каждое из которых — значение одного разряда P -ичного числа из уникальной комбинации чисел описываемого автомата. Значения разрядов P -ичного числа задаются как числа в десятичной записи через пробел. Суммарный размер входного файла не превосходит 40 000 байт.

Формат выходных данных

Ответ по каждому автомату должен содержаться в отдельной строке. Ответ — это число 0, если игроку вообще не удастся выиграть. Если же выигрыш возможен, то ответ — это число 1 и далее — M чисел через пробел в этой же строке файла: k_1, k_2, \dots, k_M , где значение k_i ($k_i \leq P$) указывает, сколько раз нужно прибавить к счетчику число a_i , чтобы в результате всех сложений счетчик обнулится. Если решение не единственно, то выведите любое из них.

Пример

arcade.in	arcade.out
3	0
4 2 2	1 1 0 0 2
2 2	0
2 2	
3 3	
3 2 4	
1 0	
2 0	
0 0	
0 1	
2 1	
14 2 2	
12 12	
10 10	
3 3	

Задача G. Векторы [1 sec, 256 mb]

Задано m векторов из нулей и единиц, длина каждого вектора равна n . Сложение векторов осуществляется покомпонентно по модулю 2.

Требуется для каждого вектора определить, можно ли его получить, сложив некоторое подмножество предыдущих, а для последнего вектора определить, какие вектора следует сложить, чтобы его получить.

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m — длина векторов и количество векторов ($1 \leq n, m \leq 2000$). Следующие m строк содержат по n целых чисел, каждое из которых равно 0 или 1.

Формат выходных данных

Для каждого вектора выведите “Yes” или “No” на отдельной строке. Если ответ для последнего вектора “Yes”, выведите $m - 1$ число, каждое из которых должно быть равно 0 или 1. Выведите 1 для тех векторов, которые следует сложить, чтобы получить последний вектор.

Пример

vectors.in	vectors.out
4 6	Yes
0 0 0 0	No
1 0 0 1	No
1 0 1 0	Yes
0 0 1 1	No
0 1 0 0	Yes
0 1 1 1	0 1 1 0 1

Задача Н. Изоморфизм графов [1 сек, 256 mb]

Вам дана матрица смежности a некоторого графа G и матрица смежности b некоторого графа G' изоморфного графу G . Оба графа неориентированные, без петель и кратных рёбер. Найдите такую перестановку p , что $a[i, j] = b[p_i, p_j]$.

Формат входных данных

На первой строке количество тестов t ($1 \leq t \leq 300$). Далее следуют t тестов, описание каждого дано следующим образом. На первой строке число n ($1 \leq n \leq 300$). Следующие n строк задают матрицу a , а следующие n строк матрицу b . Обе матрицы симметричны, на диагонали нули. Сумма n по всем тестам не превосходит 300.

Формат выходных данных

Для каждого теста выведите n чисел от 1 до n – требуемую перестановку p . Если ответов несколько, выведите любой.

Примеры

iso.in	iso.out
3	1 2
2	2 1 3
01	2 3 1
10	
01	
10	
3	
010	
101	
010	
011	
100	
100	
3	
010	
001	
000	
000	
001	
100	

Задача I. Шалтай-Болтай [0.3 сек, 256 mb]

Варкалось. Хливкие шорьки
Пырялись по наве,
И хрюкотали зелюки,
Как мюмзики в мове.

Шалтай-Болтай – существо непредсказуемое. Только он помог Алисе понять стихотворение про Бармаглота, как тут же убежал гоняться за зелюками. С тех пор как Алиса встрети­лась с ним на клетке d6, она успела дойти до восьмой горизонтали и стать королевой, а Шалтая-Болтая так никто и не видел. С момента их встречи прошло уже $100^{100^{100}}$ дней (а может, и лет – время в Зазеркалье идёт очень быстро). Определите вероятности нахождения Шалтая-Болтая на клетках Зазеркалья. Известно, что Шалтай-Болтай каждую секунду перебежал с той клетки, где он находился, на одну из соседних (соседними считаются клетки, имеющие общую вершину). Вероятность перемещения Шалтая-Болтая на ту или иную клетку прямо пропорциональна количеству зелюков на ней.

Формат входных данных

Входные данные представляют собой восемь строк, по восемь целых чисел в каждой, и определяют количество зелюков на клетках Зазеркалья. Первая строка описывает первую горизонталь (клетки с a1 по h1), а последняя строка – последнюю (клетки с a8 по h8). В каждой клетке Зазеркалья находится не менее одного и не более 1000 зелюков.

Формат выходных данных

Выведите восемь строк, по восемь вещественных чисел в каждой – вероятности нахождения Шалтая-Болтая на клетках Зазеркалья. Клетки нужно описывать в том же порядке, в котором они заданы на входе. Числа нужно выводить с точностью не менее 10^{-12} .

Примеры

humpty.in	humpty.out
1 1 1 1 1 1 1 1	0.007 0.012 0.012 0.012 0.012 0.012 0.012 0.007
1 1 1 1 1 1 1 1	0.012 0.019 0.019 0.019 0.019 0.019 0.019 0.012
1 1 1 1 1 1 1 1	0.012 0.019 0.019 0.019 0.019 0.019 0.019 0.012
1 1 1 1 1 1 1 1	0.012 0.019 0.019 0.019 0.019 0.019 0.019 0.012
1 1 1 1 1 1 1 1	0.012 0.019 0.019 0.019 0.019 0.019 0.019 0.012
1 1 1 1 1 1 1 1	0.012 0.019 0.019 0.019 0.019 0.019 0.019 0.012
1 1 1 1 1 1 1 1	0.012 0.019 0.019 0.019 0.019 0.019 0.019 0.012
1 1 1 1 1 1 1 1	0.007 0.012 0.012 0.012 0.012 0.012 0.012 0.007