

Содержание

Must have	2
Задача 11А. Определитель [0.2 sec, 256 mb]	2
Задача 11В. Линейные уравнения [0.2 sec, 256 mb]	3
Задача 11С. Обратная матрица [0.3 sec, 256 mb]	4
Обязательные задачи	5
Задача 11D. Головоломка [0.2 sec, 256 mb]	5
Задача 11Е. Никифор [0.3 sec, 256 mb]	6
Задача 11F. Векторы [0.7 sec, 256 mb]	7
Дополнительные задачи	8
Задача 11G. Разложение матрицы [1.2 sec, 256 mb]	8
Задача 11Н. Шалтай-Болтай [0.2 sec, 256 mb]	9

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же `stdin`), вывести ответ нужно в **стандартный поток вывода** (он же `stdout`).

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

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

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Must have

Задача 11А. Определитель [0.2 сек, 256 mb]

Дана матрица $n \times n$. Найти определитель.

Формат входных данных

Мультитест – от 1 до 100 тестов. Каждый тест имеет следующий вид.

n ($1 \leq n \leq 5$).

n строк по n целых чисел от 0 до 2.

Формат выходных данных

Для каждого теста одно целое число – определитель.

Примеры

stdin	stdout
2	3
2 1	0
1 2	
3	
1 0 1	
0 1 0	
1 0 1	

Замечание

Гаусс с вычислениями в типе double отлично проходит все тесты.

Задача 11В. Линейные уравнения [0.2 сек, 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

stdin	stdout
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

Если у вас WA 34, это проблемы с точностью.

Задача 11С. Обратная матрица [0.3 сек, 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

stdin	stdout
2 01 11	YES 11 10
2 00 10	NO

Подсказка по решению

bitset!

Обязательные задачи

Задача 11D. 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, пронумерованных 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

stdin	stdout
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	

Подсказка по решению

Гаусс по модулю 2.

Задача 11Е. Никифор [0.3 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

stdin	stdout
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	

Подсказка по решению

sort!

Задача 11F. Векторы [0.7 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

stdin	stdout
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

Подсказка по решению

bitset!

Дополнительные задачи

Задача 11G. Разложение матрицы [1.2 sec, 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

stdin	stdout
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 11Н. Шалтай-Болтай [0.2 sec, 256 mb]

Варкалось. Хливкие шорьки
Пырлялись 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

stdin	stdout
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

Подсказка по решению

Метод итераций?