

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. Линейные уравнения по модулю

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

Система линейных уравнений по простому модулю $p = 998244353$, как всем известно, есть множество уравнений

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

Ваша задача — решить её.

Формат входных данных

В первой строке входного файла записано одно 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 от 0 до 998244352 включительно.

Формат выходных данных

Первая строка выходного файла должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `multiple` — больше одного решения.
- `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

стандартный ввод	стандартный вывод
3 1 0 0 1 0 1 0 2 0 0 1 3	single 1 2 3
3 3 0 1 2 1 2 2 0 4 2 3 2	multiple
3 3 1 4 4 0 1 3 3 3 2 7 11	impossible

Задача С. 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 ($1 \leq N \leq 14$, $1 \leq M \leq 14$). 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

<code>traflights.in</code>	<code>traflights.out</code>
3 2	2
2 1	1 2
3 1	3 2
2 1	

Задача D. $AX = B$

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вам даны две матрицы, A (размера $n \times m$) и B (размера $n \times k$).
Найдите матрицу X (размера $m \times k$), такую, что

$$A \cdot X = B$$

то есть

$$\sum_{j=1}^m a_{i,j} x_{j,k} = b_{i,k}$$

В данной задаче произведение матриц считается по простому модулю 1 000 000 007.

Формат входных данных

В первой строке даны три целых числа — n, m, k ($1 \leq n, m, k \leq 400$).

В следующих n строках задана матрица A , в i -й строке через пробел введены $a_{i,1}, \dots, a_{i,m}$.

В следующих n строках задана матрица B , в i -й строке через пробел введены $b_{i,1}, \dots, b_{i,k}$.

Все элементы матриц — целые числа из $[0, 10^9 + 7)$.

Формат выходных данных

Выведите m строк. В i -й строке через пробел выведите k чисел — элементы матрицы X . Если возможных ответов несколько, выведите любой. Элементы должны быть целыми числами из $[0, 10^9 + 7)$.

Если подходящей матрицы X не существует, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
2 2 2 1 2 3 5 1 0 0 1	1000000002 2 3 1000000006
2 3 3 1 2 3 2 3 1 8 10 8 9 9 9	1 2 1 2 1 2 1 2 1
1 1 2 10 2 5	4000000003 5000000004
2 1 1 1 3 2 1	-1

Замечание

В первом примере нужно найти обратную к квадратной матрице. $\begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix} \cdot \begin{pmatrix} -5 & 2 \\ 3 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

В третьем примере искомая матрица над полем вещественных чисел равна $\begin{pmatrix} \frac{1}{5} & \frac{1}{2} \end{pmatrix}$.

Задача Е. 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 F. Мышь в лабиринте

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

Задано поле размером $N \times M$ клеток. Каждая его клетка либо пуста, либо непроходима (содержит стенку), либо содержит сыр.

В одной из пустых клеток поля изначально находится мышь. Каждый ход мышь выбирает из текущей клетки и свободных соседних (по стороне или углу) клеток с равной вероятностью и переходит в неё. Например, если мышь стоит не с краю и все соседние клетки свободны, то с вероятностью $\frac{1}{9}$ она останется на месте, или перейдёт в одну из соседних клеток с вероятностью $\frac{1}{9}$ в каждую.

Если мышь попадает в клетку с сыром, то она съедает его, и процесс заканчивается.

Требуется найти математическое ожидание числа сделанных ходов.

Формат входных данных

В первой строке записаны числа N и M — размеры поля ($1 \leq N, M \leq 20$).

Далее идут N строк по M символов, содержащие описание поля. Символ «.» обозначает пустую клетку, «#» — стенку, «С» — сыр, «М» — клетку с мышью. Гарантируется, что есть ровно одна клетка с символом «М».

Формат выходных данных

Выведите искомое математическое ожидание числа ходов или «-1», если съесть сыр невозможно. Абсолютная или относительная погрешность должна быть не хуже 10^{-4} .

Примеры

стандартный ввод	стандартный вывод
1 2 МС	2.0000000000
1 3 М.С	7.0000000000
3 3 ### МС# #. #	3.0000000000
2 3 М#С .#.	-1
5 5 ССССС .###. ..М.# .#### #С.#.	10.8195266272

Задача G. 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 Н. Число возможных векторов

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

Дан набор из m битовых векторов размера n ($1 \leq n, m \leq 50$). Сколько различных векторов можно получить как хог этих векторов.

Формат входных данных

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

Формат выходных данных

Выведите число векторов, которые можно получить

Примеры

стандартный ввод	стандартный вывод
2 2 10 11	4
3 4 100 111 011 010	8
3 4 111 111 111 000	2

Задача I. 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 $\wedge$, 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 элементов массива: $X(A) = A_1 \oplus A_2 \oplus \dots \oplus A_n$. Аналогично, введем обозначение $X(B) = B_1 \oplus B_2 \oplus \dots \oplus B_n$.

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 ($1 \leq n \leq 10^5$). В следующей строке находится n элементов массива A ($0 \leq A_i \leq 10^{18}$). 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	6 1
1 1	1
2 2	

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 J. Полные квадраты

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

Множество целых положительных чисел будем называть *полноквадратным*, если произведение его элементов является полным квадратом (равно 1, 4, 9, 16, 25, 36, ...).

Задано множество A . Определите, сколько непустых подмножеств B множества A являются полноквадратными.

Формат входных данных

В первой строке записано целое число N — количество элементов множества ($1 \leq N \leq 100$). Во второй строке записаны N попарно различных чисел a_i — элементы множества ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно число — количество полноквадратных подмножеств по модулю 1 000 000 007.

Примеры

<code>perfect-squares.in</code>	<code>perfect-squares.out</code>
4 49 20 500 7	3

Задача К. 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

Задача L. Совершенство

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

Павел Андреевич очень любит все совершенное. Его новая любовь - совершенные паросочетания! Не огорчайте Павла Андреевича, проверьте, есть ли в данном графе совершенное паросочетание.

Формат входных данных

В первой строке даны два числа N и M - количество вершин и количество ребер в G . $N \leq 100$. Следующие M строк содержат числа a_i и b_i - ребра графа. Гарантируется отсутствие петель и кратных ребер.

Формат выходных данных

Выведите слово «YES» или слово «NO» - ответ на поставленную задачу.

Примеры

<code>matchme.in</code>	<code>matchme.out</code>
6 7 1 2 2 3 1 3 5 6 6 4 4 5 6 2	YES
3 3 1 2 2 3 1 3	NO

Задача М. Число остовных деревьев

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

Дан связный неориентированный граф. Найдите число его остовных деревьев по модулю $10^9 + 7$.

Формат входных данных

Первая строка входного файла содержит натуральные числа n и m — количество вершин и ребер графа ($1 \leq n \leq 100$, $1 \leq m \leq 1000$). Далее следуют m строк, задающих ребра. Граф не содержит кратных ребер и петель.

Формат выходных данных

Выведите число остовных деревьев по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
4 4 2 1 4 2 2 3 4 3	3

Задача N. Электричество в каждый дом!

Имя входного файла:	<code>countspans.in</code>
Имя выходного файла:	<code>countspans.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Известный чешский математик Отакар Боровка крайне увлечен проектированием электросети Моравии. В Моравии N городов, и различные строительные фирмы уже предложили Отакару M проектов построения ЛЭП между какими-то двумя городами. Известно, что в погоне за индивидуальностью и неповторимым строительным почерком, каждая фирма предлагает все свои проекты по одинаковой стоимости, отличной от стоимостей проектов других фирм. Также известно, что каждая фирма предлагает не более *трех* проектов. Можете считать, что фирмы достаточно сообразительны, чтобы не предлагать проектов соединения какого то города с самим собой, но вполне может возникнуть ситуация, что одна или несколько фирм предлагают больше одного проекта соединения одной и той же пары городов.

Боровка собрал всех своих друзей и поручил им задачу спроектировать электросеть минимальной стоимости. Как вы, наверное, уже догадались, электрическая сеть является остовным деревом, а Боровку интересуют только сети, стоимость постройки которых минимальна.

Боровка всегда был уверен, что минимальное остовное дерево у графа одно, и представьте себе его удивление, когда каждый из друзей принес ему свой проект, утверждая что его-то дерево и есть минимальное. Подозревая неладное, он думает, что причиной разных ответов стали ребра одинакового веса. Помогите ему — посчитайте количество возможных электросетей минимальной стоимости, состоящих из ЛЭП, предложенных Боровке.

Формат входных данных

В первой строке входного файла заданы два целых положительных числа N и M , не превосходящие 100 000.

Следующие M строк содержат по 3 целых числа каждая: $1 \leq a_i, b_i \leq N$ и $1 \leq c_i \leq 10^9$ — города, соединенные соответствующей ЛЭП, и ее стоимость. Гарантируется, что для любого числа c_i найдется не более трех ЛЭП, имеющих такую стоимость.

Формат выходных данных

Выведите одно число — ответ на задачу. Так как это число может оказаться довольно большим, выведите остаток от деления на $10^9 + 7$.

Примеры

countspans.in	countspans.out
2 2 1 2 1 2 1 1	2
3 5 1 2 2 2 3 3 3 1 3 3 1 1 2 3 1	1
5 10 2 5 8 4 5 8 2 4 3 4 2 3 4 2 3 5 4 6 5 1 6 1 3 5 3 1 5 4 5 7	6

Задача О. Ксор-базис

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

Вам онлайн приходят неотрицательные целые числа, каждое строго меньше 2^m . В любой момент, обозначив за S множество уже полученных чисел, назовём число **представимым**, если его можно получить как хог некоторого подмножества S (пустое подмножество даёт 0). Множество всех представимых чисел обозначим $X(S)$; оно всегда содержит 0, а разные подмножества S могут давать одинаковый хог, поэтому в $X(S)$ учитываются только ***различные*** значения.

Нужно в онлайн обрабатывать запросы трёх типов:

- ***добавить*** в S новое число v ; - по данному x найти ***количество*** различных представимых чисел, не превосходящих x ; - по данному k найти ***k-е по возрастанию*** (нумерация с единицы) различное представимое число.

Все запросы обрабатываются в порядке поступления: ответ на запрос нужно вывести, не зная будущих запросов.

Формат входных данных

В первой строке два целых числа n и m ($1 \leq n \leq 5000$, $1 \leq m \leq 2000$) — число запросов и длина чисел в битах.

В следующих n строках записаны запросы. Каждое число задаётся своей ***двоичной*** записью старшим битом вперёд, без ведущих нулей (число 0 записывается как '0'). Запрос имеет один из видов:

- '1 B' — добавить в S число v с двоичной записью B ($0 \leq v < 2^m$, длина B не превосходит m); - '2 B' — вывести количество различных представимых чисел, не превосходящих x , где x имеет двоичную запись B ($0 \leq x < 2^m$, длина $B \leq m$); - '3 B' — вывести k -е по возрастанию различное представимое число, где k имеет двоичную запись B ($1 \leq k \leq 2^m$, длина $B \leq m + 1$).

Формат выходных данных

На каждый запрос типа '2' выведите в отдельной строке искомое количество в ***двоичной*** записи (без ведущих нулей).

На каждый запрос типа '3' выведите в отдельной строке искомое число в двоичной записи, либо '-1', если различных представимых чисел меньше k .

Примеры

стандартный ввод	стандартный вывод
7 5 1 10101 1 1100 1 10011 2 11111 3 110 3 1000 2 0	1000 10101 11111 1
4 3 2 111 3 1 3 10 1 0	1 0 -1
6 4 1 11 1 101 2 101 3 100 2 110 3 101	11 110 100 -1