

Задача А. 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 входного файла

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 выходного файла

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

<code>quadratic.in</code>	<code>quadratic.out</code>
0 1	1 1 0
2 1 1 0	
3 1 0 0 0	

Задача В. Связность графа

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

Дан связный неориентированный граф. Вам поступают запросы вида: проверить, останется ли граф связным после удаления некоторого маленького множества ребер.

Формат входного файла

Первая строка входного файла содержит два числа — N и M ($1 \leq N \leq 10000$, $1 \leq M \leq 100000$), обозначающие число вершин и число ребер, соответственно. Следующие M строк содержат описание ребер. Каждая строка состоит из двух чисел a и b — номера вершин, соединяемых соответствующим ребром. В графе нет петель и кратных ребер. Вершины графа нумеруются с единицы. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Следующая строка содержит единственное число K ($1 \leq K \leq 100000$), обозначающее число запросов. Следующие K строк содержат описания запросов. Каждое описание начинается с числа C ($1 \leq C \leq 4$), обозначающее число ребер в запросе, далее следуют C чисел, обозначающих номера ребер, входящих в запрос. Все ребра, входящие в запрос, являются различными.

Формат выходного файла

Для каждого запроса выведите единственную строку. В i -ой строке должно содержаться слово «Connected», если удаление всех ребер из соответствующего сохранит связность графа, и «Disconnected» в противном случае.

Примеры

<code>disconnected.in</code>	<code>disconnected.out</code>
4 5	Connected
1 2	Disconnected
2 3	Connected
3 4	
4 1	
2 4	
3	
1 5	
2 2 3	
2 1 2	

Задача С. 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 входного файла

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 выходного файла

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 D. Оксюморон

Имя входного файла: product.in
Имя выходного файла: product.out
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Даны три квадратные матрицы одинакового размера. Проверьте, что третья матрица является произведением первой на вторую (в таком порядке!).

Формат входного файла

Входной файл состоит из одного или нескольких тестовых примеров. Каждый тестовый пример начинается с одного числа N — размера матриц ($1 \leq N \leq 500$). Далее следует описание трех матриц, каждая матрица описывается N строками по N целых чисел каждая. Все числа во входного файле не превосходят 10^9 . Файл завершается примером с $N = 0$, его обрабатывать не требуется.

Формат выходного файла

Выведите YES, если третья матрица является произведением первой на вторую, и NO в противном случае.

Примеры

product.in	product.out
2 1 0 0 1 1 0 0 1 1 0 0 1 2 1 0 0 1 1 0 0 1 1 0 0 0 0	YES NO

Задача E. Звезда смерти-2

Имя входного файла: star2.in
Имя выходного файла: star2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Давным-давно в далекой-предалекой галактике...

Боевая космическая станция «Звезда Смерти» была спроектирована еще до начала Клонических войн. Через много лет она была передана в руки Империи для контроля над Внешними Территориями. «Звезда Смерти» имела более 100 миль в диаметре, была оснащена гравитонной пушкой, способной уничтожать целые планеты, а также могла нести на борту несколько тысяч истребителей. «Звезда Смерти» должна была наводить ужас на население и полностью исключить всякую возможность сопротивления властям Империи.

После того, как первая «Звезда Смерти» была уничтожена повстанцами, началось создание новой модели, еще более смертоносной. Новая модель, как и первая, имеет форму шара и может поступательно перемещаться в N -мерном пространстве. Она оснащена M жестко закрепленными криптоновыми двигателями. Если на i -й двигатель подать X единиц энергии, то вклад этого двигателя в j -ю координату вектора тяги составит $A_{ij} \cdot X$. В зависимости от режима работы, криптоновый двигатель может перемещать станцию вперед либо назад (в этом случае X отрицательно). Итоговый вектор тяги равняется сумме вкладов каждого из M двигателей.

Перед началом движения станции специальный навигационный модуль определяет необходимые координаты вектора тяги (b_1, b_2, \dots, b_N) . Ваша программа должна вычислить количество энергии, которое необходимо подать на каждый из двигателей так, чтобы длина вектора разности суммарной тяги и необходимой тяги была минимальна. В слу-

чае, если ответ неоднозначен, требуется дополнительно минимизировать сумму квадратов величин энергии, подаваемой на двигатели.

Формат входного файла

В первой строке через пробел записаны числа N и M ($1 \leq N, M \leq 100$). Далее в M строках по N чисел в строке следует матрица A_{ij} . В последней строке записаны N чисел b_j — координаты необходимого вектора тяги. Все числа A_{ij} и b_j целые и по модулю не превосходят 100.

Формат выходного файла

Выведите M действительных чисел X_1, \dots, X_M с точностью в пять знаков после десятичной точки. X_i должно равняться количеству энергии, которое необходимо подать на i -й двигатель. В случае, если ответов несколько, выведите любой.

Примеры

star2.in	star2.out
4 3 2 3 -2 1 -1 2 1 3 4 2 3 -2 3 13 -9 13	4.00000 2.00000 -1.00000

Задача F. Very simple problem

Имя входного файла: simplex.in
Имя выходного файла: simplex.out
Ограничение по времени: 10 секунды
Ограничение по памяти: 256 мегабайт

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \\ x_1, \dots, x_n \geq 0 \end{cases}$$
$$c_1x_1 + \dots + c_nx_n \rightarrow \max$$

Формат входного файла

В первой строк входных данных содержится два целых числа: n и m — количество переменных и количество уравнений, соответственно ($1 \leq n \leq 600, 1 \leq m \leq 1000$). Следующие m строк содержат описания уравнений, описание i -ого уравнения содержит вещественные числа a_{i1}, \dots, a_{in} и b_i . Последняя строка содержит n вещественных чисел: c_1, \dots, c_n . Все a_{ij} не превосходят 10^3 по модулю. Все b_i и c_i не превосходят 10^9 по модулю.

Формат выходного файла

В случае, если данная задача не имеет допустимых решений, выведите **No solution**. Если же можно получить неограниченно большое значение целевой функции, выведите **Unbounded**.

В противном случае в первой строке выведите слово **Bounded**. Во второй строке выведите значения переменных, для которых достигается максимум целевой функции. Ваш ответ будет считаться правильным, если значение целевой функции в вашей точке будет отличаться от правильного не более, чем на 10^{-6} по абсолютной или относительной погрешности.

Примеры

simplex.in	simplex.out
2 2 1 2 3 2 1 3 1 1	Bounded 1.0000000000 1.0000000000
2 1 -1 -1 0 1 1	Unbounded
2 1 1 1 -1 1 1	No solution