

Задача А. Расстояние между отрезками в 3D

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

Вам даны два отрезка в трехмерном пространстве. Требуется найти расстояние между этими отрезками. Расстоянием между отрезками называется минимальное расстояние среди всех пар точек, принадлежащим этим отрезкам.

Формат входных данных

В первой строке записано три целых числа, разделенных пробелами — координаты одного из концов первого отрезка, в следующей строке так же записаны три целых числа — координаты другого отрезка.

Во третьей и четвертой строках аналогичным образом записано описание второго отрезка.

Все координаты не превосходят по модулю 1000.

Формат выходных данных

Выведите одно вещественное число — расстояние между отрезками. Ответ будет считаться верным, если абсолютная или относительная погрешность не будет превосходить 10^{-6} .

Примеры

<code>dist.in</code>	<code>dist.out</code>
0 0 0 1 0 0 0 2 0 0 2 1	2.0000000000
2 1 0 0 1 0 0 0 0 3 3 0	0.0000000000

Задача В. Корень кубического уравнения

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

Дано кубическое уравнение $ax^3 + bx^2 + cx + d = 0$ ($a \neq 0$). Известно, что у этого уравнения есть ровно один корень. Требуется его найти.

Формат входных данных

Во входном файле через пробел записаны четыре целых числа: $-1000 \leq a, b, c, d \leq 1000$.

Формат выходных данных

Выведите единственный корень уравнения с точностью не менее 5 знаков после десятичной точки.

Примеры

стандартный ввод	стандартный вывод
1 -3 3 -1	1
-1 -6 -12 -7	-1.000000

Задача С. Три круга

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

Даны $N \leq 3$ круга на плоскости. Посчитайте площадь объединения.

Данную версию задачи предлагается решать методом Монте-Карло.

Формат входных данных

В первой строке число N ($1 \leq N \leq 3$).

В следующих N строках содержится по 3 целых числа — координаты центра и положительный радиус.

Все круги целиком лежат внутри квадрата $(0; 0) \times (100; 100)$.

Формат выходных данных

Выведите одно вещественное число — площадь объединения.

Допустимая относительная погрешность — 10^{-3} .

Примеры

стандартный ввод	стандартный вывод
1 99 99 1	3.14159325171203818172
2 2 2 2 1 1 1	13.15189494562218897045

Задача D. Ферзи

Имя входного файла: `queens.in`
Имя выходного файла: `queens.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

На доске $N \times N$ расставьте, пожалуйста, N обычных шахматных ферзей так, чтобы они друг друга не били.

Формат входных данных

В единственной строке входных данных содержится одно целое число N — размер доски ($4 \leq N \leq 200$).

Формат выходных данных

Для каждой горизонтали исходной доски выведите номер вертикали, на которой стоит ферзь в этой горизонтали, вертикали нумеруются слева направо, начиная с единицы.

Примеры

<code>queens.in</code>	<code>queens.out</code>
4	2 4 1 3

Задача Е. Гориллы против ЛКШат

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

В Горилляндии есть n городов, каждые два из которых соединены дорогой. Эти дороги были построены в давние времена *восточными* и *западными* гориллами. Дороги, которые были построены *восточными* гориллами, вымощены белыми камнями, а те, что построены *западными* — черными. Поскольку гориллы уже изобрели политическую дискриминацию, ни одна *восточная* горилла не может пройти по дороге, вымощенной черными камнями, и ни одна *западная* — по белой дороге.

Когда-то давно ЛКШата решили избрать своих правителей и изгнали горилл из Горилляндии. Однако недавно *восточные* и *западные* гориллы договорились вернуть Горилляндию под свой контроль. Для этого они хотят направить в некоторые города Горилляндии горилл, которые возьмут эти и смежные с ними города под свой контроль.

Точнее, если *восточный* горилл будет направлен в некоторый город, то он возьмет под свой контроль этот город и все города, которые напрямую соединены с ним белыми дорогами. Аналогично, *западный* горилл помимо города, в который он направлен, будет контролировать все города, напрямую соединенные с ним черными дорогами. Для захвата Горилляндии требуется установить контроль над всеми городами.

Однако, при разработке плана захвата обнаружилось две трудности. Во-первых, выяснилось, что горилл согласен принять участие в операции только если все гориллы, которые будут направлены в Горилляндию, будут представлять тот же вид, что и он. То есть либо все участвующие в захвате гориллы должны быть *восточными*, либо все они должны быть *западными*. Во-вторых, общее число горилл, которые могут быть направлены в Горилляндию не должно превышать k . Единственная надежда горилл заключается в том, что k достаточно велико, $2^k \geq n$.

Выясните, *восточных* или *западных* горилл следует использовать для захвата Горилляндии, а также в какие города их следует направить.

Формат входных данных

В первой строке вводятся целые числа n и k ($2 \leq n \leq 256$, $2^k \geq n$, $k \leq n$).

Следующие n строк содержат по n целых чисел каждая. На i -й позиции i -й из этих строк расположено число 0, которое означает, что город не соединен дорогой сам с собой. Для всех $j \neq i$ число на j -й позиции i -й из этих строк равно 1, если i -й город соединен с j -м белой дорогой, и равно 2, если они соединены черной дорогой. Числа в строках разделены пробелами.

Гарантируется, что входные данные корректны, то есть если i -й город соединен с j -м белой дорогой, то и j -й соединен с i -м белой дорогой, аналогично в случае черных дорог.

Формат выходных данных

Если захватить Горилляндию при заданных условиях невозможно, выведите единственное число 0. В противном случае в первой строке выведите 1, если удастся захватить Горилляндию с использованием восточных горилл, и 2, если требуется использовать западных горилл.

В следующей строке выведите число $l \leq k$ — количество использованных горилл.

Третья строка должна содержать l целых чисел — номера городов, в которые следует направить горилл.

Заметьте, что вам не требуется минимизировать l . Если решений несколько, выведите любое.

Пример

стандартный ввод	стандартный вывод
8 3	1
0 1 1 1 1 2 2 2	3
1 0 1 1 2 1 2 2	2 5 6
1 1 0 1 2 2 1 2	
1 1 1 0 2 2 2 1	
1 2 2 2 0 2 1 1	
2 1 2 2 2 0 2 1	
2 2 1 2 1 2 0 2	
2 2 2 1 1 1 2 0	

Задача F. Хорошие раскраски

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

Назовем раскраску клеток таблицы $n \times m$ *хорошей*, если никакие четыре клетки, центры которых образуют вершины прямоугольника со сторонами, параллельными осям координат, не покрашены в один цвет.

Иначе говоря, для раскраски не должно быть четверки целых чисел x_1, x_2, y_1, y_2 , что $1 \leq x_1 < x_2 \leq n$, $1 \leq y_1 < y_2 \leq m$, и клетки (x_1, y_1) , (x_2, y_1) , (x_1, y_2) и (x_2, y_2) покрашены в одинаковый цвет.

Требуется написать программу, которая по заданным целым числам n , m и c находит любую хорошую раскраску таблицы $n \times m$ в c цветов.

Формат входных данных

В первой строке записаны три целых числа n, m, c ($2 \leq n, m \leq 10$, $2 \leq c \leq 3$).

Гарантируется, что для заданных во входных данных значений существует хотя бы одна хорошая раскраска.

Формат выходных данных

Выведите n строк по m чисел в каждой.

В качестве j -го числа i -й строки выведите $a_{i,j}$ — цвет клетки (i, j) ($1 \leq a_{i,j} \leq c$).

Если есть несколько хороших раскрасок, можно вывести любую из них.

Пример

стандартный ввод	стандартный вывод
2 2 2	1 2 2 2

Задача G. ЮграНефтеТранс

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайта

Ханты-Мансийский автономный округ — Югра является важнейшим нефтяным регионом России. Добыча нефти составляет 267 млн т в год, её транспортировка осуществляется по трубопроводам, общая длина которых превышает длину экватора Земли.

Система транспортировки нефти представляет собой совокупность n распределительных станций и m трубопроводов. Каждый трубопровод соединяет две различные станции. Между любыми двумя станциями проложено не более одного трубопровода.

Эффективность работы станций существенно зависит от вязкости нефти. Поэтому компания «ЮграНефтеТранс», в ведении которой находится сеть трубопроводов, заказала инновационному исследовательскому предприятию разработку и изготовление новых сверхточных датчиков вязкости на основе самых современных технологий.

Изготовление датчиков — процесс трудоёмкий и дорогостоящий, поэтому было решено изготовить k датчиков ($k \leq 40$) и выбрать k различных станций, на которых датчики будут установлены. Необходимо осуществить выбор станций так, чтобы датчики контролировали все трубопроводы: для каждого трубопровода хотя бы один датчик должен быть установлен на станции, где начинается или заканчивается этот трубопровод.

Напишите программу, которая проверяет, существует ли требуемое расположение датчиков, и в случае положительного ответа находит это расположение.

Формат входных данных

В первой строке входного файла записаны три натуральных числа — n , m и k ($k \leq n \leq 2000$, $1 \leq m \leq 10^5$, $1 \leq k \leq 40$). Далее следуют m строк, каждая из которых описывает один трубопровод. Трубопровод задаётся двумя целыми числами — порядковыми номерами станций, которые он соединяет. Станции пронумерованы от 1 до n . Гарантируется, что к любой станции подведён хотя бы один трубопровод и между любыми двумя станциями проложено не более одного трубопровода. Числа в каждой строке разделены пробелами.

Формат выходных данных

В первую строку выходного файла выведите слово **Yes**, если требуемое расположение датчиков существует, в противном случае — слово **No**. В случае положительного ответа выведите во вторую строку выходного файла k различных целых чисел — номера станций, на которых необходимо установить датчики. Номера можно выводить в любом порядке. Если существует несколько подходящих расположений датчиков, выведите любое из них. Разделяйте числа во второй строке пробелами.

Примеры

stdin	stdout
2 1 2 1 2	Yes 2 1
3 3 1 1 2 2 3 3 1	No
7 6 2 1 2 1 3 1 4 2 5 2 6 2 7	Yes 1 2
5 5 2 1 2 1 3 1 4 1 5 4 5	Yes 4 1