Задача A. Петли

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

По заданной матрице смежности неориентированного графа определите, содержит ли он петли.

Формат входных данных

В первой строке входного файла дано число N ($1 \leqslant N \leqslant 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл вывести «YES», если граф содержит петли, и «NO» в противном случае.

Примеры

loops.in	loops.out
3 0 1 1 1 0 1 1 1 0	NO
3 0 1 0 1 1 1 0 1 0	YES

Задача B. Количество ребер в неориентированном графе

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Простой неориентированный граф задан матрицей смежности. Найти количество ребер в графе.

Формат входных данных

В первой строке входного файла дано число N ($1 \leqslant N \leqslant 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл выведите единственное число — количество ребер в графе.

Примеры

edges.in	edges.out
3 0 1 0 1 0 1 0 1 0	2

Задача C. Проверка на неориентированность

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

По заданной квадратной матрице $N \times N$ из нулей и единиц определить, может ли данная матрица быть матрицей смежности простого неориентированного графа.

Формат входных данных

В первой строке входного файла дано число N ($1 \leqslant N \leqslant 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл вывести «YES», если граф неориентированный, и «NO» в противном случае.

Примеры

orient.in	orient.out
3 0 1 1 1 0 1 1 1 0	YES
3 0 1 1 1 0 1 0 1 0	NO

Задача D. Степени вершин

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Простой неориентированный граф задан матрицей смежности. Найдите степени всех вершин графа.

Формат входных данных

В первой строке входного файла дано число N ($1 \leqslant N \leqslant 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В выходной файл выведите N чисел — степени всех вершин.

Примеры

vertexes.in	vertexes.out
3 0 1 0 1 0 1 0 1 0	1 2 1

Задача E. Истоки и стоки

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вершина ориентированного графа называется истоком, если в нее не входит ни одно ребро, стоком, если из нее не выходит ни одного ребра. Ориентированный граф задан матрицей смежности. Найдите все вершины графа, которые являются истоками и все его вершины, которые являются стоками.

Формат входных данных

В первой строке входного файла дано число N ($1 \leqslant N \leqslant 100$). Затем идут N строк по N элементов в каждой — описание матрицы смежности.

Формат выходных данных

В первой строке выходного файла выведите число k — число истоков в графе и затем k чисел — номера вершин, которые являются истоками, в возрастающем порядке. На второй строке выведите информацию о стоках в том же формате.

Примеры

flow.in	flow.out
4 1 0 0 1 0 0 0 0 1 1 0 1 0 0 0 0	1 3 2 2 4

Задача F. Издевательство

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Штирлиц ехал на машине, увидел голосующего Бормана, и проехал мимо. Через некоторое время он снова увидел голосующего Бормана, и снова проехал мимо. Вскоре он опять увидел голосующего Бормана.
— Издевается! — подумал Борман.
— Кольцевая! — догадался Штирлиц.

В городе N площадей. Любые две площади соединены между собой ровно одной дорогой с двусторонним движением. В этом городе живет Штирлиц. У Штирлица есть хобби — он любит воскресным утром выйти из дома, сесть в машину, выбрать какой-нибудь кольцевой маршрут, проходящий ровно по трем площадям (то есть сначала он едет с какой-то площади на какую-то другую, потом — на третью, затем возвращается на начальную, и опять едет по этому маршруту). Он воображает, что где-то на этом пути стоит Борман. И так вот ездит Штирлиц все воскресенье, пока голова не закружится.

Естественно, что Штирлицу хочется проезжать мимо точки, в которой, как он воображает, стоит Борман, как можно чаще. Для этого, естественно, выбранный Штирлицем маршрут должен быть как можно короче. Напишите программу, которая выберет оптимальный для Штирлица маршрут.

Формат входных данных

В первой строке задается число N ($3 \leqslant N \leqslant 100$). В последующих строках содержится матрица $N \times N$ расстояний между площадями (число в позиции i,j обозначает длину дороги, соединяющей i-ую и j-ую площади). Все числа в матрице (кроме стоящих на главной диагонали) — натуральные, не превышающие 1000. Матрица симметрична относительно главной диагонали, на главной диагонали стоят 0.

Формат выходных данных

Требуется вывести номера площадей в оптимальном маршруте. Если маршрутов несколько, выведите любой из них.

Примеры

mockery.in	mockery.out
5 0 20 10 30 40 20 0 30 1 2 10 30 0 40 1000 30 1 40 0 21 40 2 1000 21 0	4 5 2

Задача G. Перекраска клеток

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Дано клетчатое поле $N \times M$, все клетки поля изначально белые. Автомат умеет:

- закрасить клетку (i, j) в черный цвет.
- для клетки (i, j) узнать её ближайших белых соседей по вертикали и горизонтали.

Дана последовательность команд для автомата. Требуется выполнить эти команды в указанной последовательности, и для каждой команды запроса ближайших белых соседей указать результат ее выполнения.

Формат входных данных

Сначала вводятся размеры поля N и M ($1 \leqslant N \leqslant 20, 1 \leqslant M \leqslant 50000$), затем количество команд K ($1 \leqslant K \leqslant 10^5$), а затем сами команды. Команды записаны по одной в строке в следующем формате:

- `Color i j` — окраска клетки (i, j) в черный цвет;
- `Neighbors i j` — нахождение белых соседей для клетки (i, j).

Клетка (i, j) в описании любой команды может быть как белой, так и черной. $1 \leqslant i \leqslant N, 1 \leqslant j \leqslant M$.

Формат выходных данных

На каждый запрос `Neighbors` требуется вывести сначала количество ближайших белых соседей (или 0, если ни с одной из сторон белых клеток не осталось), а затем их координаты (соседей можно перечислять в произвольном порядке). Если запросов `Neighbors` нет, ничего выводить не надо.

Примеры

repaint.in	repaint.out
5 5 6 Color 4 2 Neighbors 4 3 Color 2 3 Color 3 3 Neighbors 4 3 Neighbors 5 1	4 4 1 4 4 3 3 5 3 4 4 1 4 4 1 3 5 3 2 5 2 4 1