

Задача А. 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 ($1 \leq N \leq 100$). 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 В. 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 ($1 \leq N \leq 100$). 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 С. 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 ($1 \leq N \leq 100$). 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 ($1 \leq N \leq 100$). 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	1
0 1 0	2
1 0 1	1
0 1 0	

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 ($1 \leq N \leq 100$). 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 3
1 0 0 1	2 2 4
0 0 0 0	
1 1 0 1	
0 0 0 0	

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 ($3 \leq N \leq 100$). 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	4 5 2
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	

Задача G. От матрицы смежности к списку ребер

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

Простой неориентированный граф задан матрицей смежности, выведите его представление в виде списка ребер.

Формат входных данных

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

Формат выходных данных

В выходной файл выведите список ребер, упорядоченный вначале по первой вершине в паре вершин, которая описывает ребро, а затем по второй. Каждое ребро должно задаваться вначале концом с меньшим номером, а затем с концом с большим номером.

Примеры

tolist.in	tolist.out
3	1 2
0 1 1	1 3
1 0 1	2 3
1 1 0	

Задача Н. От матрицы смежности к списку ребер-2

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

Ориентированный граф задан матрицей смежности, выведите его представление в виде списка ребер.

Формат входных данных

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

Формат выходных данных

В выходной файл выведите список ребер, упорядоченный вначале по первой вершине в паре вершин, которая описывает ребро, а затем по второй.

Примеры

tolist2.in	tolist2.out
3	1 2
0 1 0	2 3
0 0 1	3 1
1 1 0	3 2

Задача I. От списка ребер к матрице смежности

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

Простой неориентированный граф задан списком ребер, выведите его представление в виде матрицы смежности.

Формат входных данных

В первой строке входного файла заданы два целых числа N ($1 \leq N \leq 100$) — число вершин и M ($1 \leq M \leq \frac{N(N-1)}{2}$) — число ребер. Далее в M строках содержатся M пар чисел, каждая из которых описывает одно ребро графа.

Формат выходных данных

В выходной файл выведите матрицу смежности графа.

Примеры

tomatrix.in	tomatrix.out
3 3	0 1 1
1 2	1 0 1
2 3	1 1 0
1 3	

Задача J. Проверка на наличие параллельных ребер

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

Неориентированный граф задан списком ребер. Проверьте, содержит ли он параллельные ребра.

Формат входных данных

Сначала вводятся числа n ($1 \leq n \leq 100$) — количество вершин в графе и m ($1 \leq m \leq 10000$) — количество ребер. Затем следует m пар чисел — ребра графа.

Формат выходных данных

Выведите YES, если граф содержит параллельные ребра, и NO в противном случае.

Примеры

parallel.in	parallel.out
3 3 1 2 2 3 1 3	NO
3 3 1 2 2 3 2 1	YES

Задача К. Полустепени вершин по спискам ребер

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

Ориентированный граф задан списком ребер. Найдите полустепени всех вершин графа.

Формат входных данных

Сначала вводятся числа n ($1 \leq n \leq 100$) — количество вершин в графе и m ($1 \leq m \leq n(n-1)$) — количество ребер. Затем следует m пар чисел — ребра графа.

Формат выходных данных

Выведите n пар чисел — для каждой вершины сначала выведите полустепень захода и затем полустепень исхода.

Примеры

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

Задача L. Обрати меня!

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

Разверните ориентированный граф.

Формат входных данных

Во входном файле записано число N ($1 \leq N \leq 50000$) — количество вершин в графе. В следующих N строках записан граф в виде списков смежности: в i -ой строке, в порядке возрастания, записаны номера вершин, в которые идут ребра из i -ой вершины. Нумерация начинается с единицы. Гарантируется, что ребер в графе не более 300000.

Формат выходных данных

Выведите развернутый граф в том же формате, что и исходный.

Примеры

reverse.in	reverse.out
4	4
2 3	1 4
3	1 2
2	

Задача М. Свинки-копилки

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

У Васи есть N свинок-копилков, свинки занумерованы числами от 1 до N . Каждая копилка может быть открыта единственным соответствующим ей ключом или разбита.

Вася положил ключи в некоторые из копилков (он помнит, какой ключ лежит в какой из копилков). Теперь Вася собрался купить машину, а для этого ему нужно достать деньги из всех копилков. При этом он хочет разбить как можно меньшее количество копилков (ведь ему еще нужно копить деньги на квартиру, дачу, вертолет...). Помогите Васе определить, какое минимальное количество копилков нужно разбить.

Формат входных данных

В первой строке содержится число N — количество свинок-копилков ($1 \leq N \leq 100000$). Далее идет N строк с описанием того, где лежит ключ от какой копилки: в i -ой из этих строк записан номер копилки, в которой находится ключ от i -ой копилки.

Формат выходных данных

Выведите единственное число: минимальное количество копилков, которые необходимо разбить.

Примеры

pigs.in	pigs.out
4	2
2	
1	
2	
4	