

Задача А. Города и дороги

Имя входного файла: `cities.in`
Имя выходного файла: `cities.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Формат входного файла

Во входном файле записано число N ($0 \leq N \leq 100$). В следующих N строках записано по N чисел, каждое из которых является единичкой или ноликом. Причем, если в позиции (i, j) квадратной матрицы стоит единичка, то i -й и j -й города соединены дорогами, а если нолик, то не соединены.

Формат выходного файла

В выходной файл выведите одно число — количество дорог в этой стране.

Пример

<code>cities.in</code>	<code>cities.out</code>
5 0 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	3

Задача В. Светофоры

Имя входного файла: `lights.in`
Имя выходного файла: `lights.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайт

В подземелье M тоннелей и N перекрестков, каждый тоннель соединяет какие-то два перекрестка. Мышиный король решил поставить по светофору в каждом тоннеле перед каждым перекрестком. Напишите программу, которая посчитает, сколько светофоров должно быть установлено на каждом из перекрестков. Перекрестки пронумерованы числами от 1 до N .

Формат входного файла

Во входном файле записано два числа N и M ($0 < N \leq 100$), $0 \leq M \leq \frac{N(N-1)}{2}$). В следующих M строках записаны по два числа i и j ($1 \leq i, j \leq N$), которые означают, что перекрестки i и j соединены тоннелем.

Формат выходного файла

В выходной файл вывести N чисел: k -е число означает количество светофоров на k -м перекрестке.

Пример

<code>lights.in</code>	<code>lights.out</code>
7 10 5 1 3 2 7 1 5 2 7 4 6 5 6 4 7 5 2 1 5 3	3 3 2 2 5 2 3

Задача С. От списка ребер к матрице смежности

Имя входного файла: `e2m.in`
Имя выходного файла: `e2m.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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

Формат входного файла

Входной файл содержит числа N ($1 \leq N \leq 100$) — число вершин в графе и M ($1 \leq M \leq \frac{n(n-1)}{2}$) — число ребер. Затем следует M пар чисел — ребра графа.

Формат выходного файла

Выведите в выходной файл матрицу смежности заданного графа.

Пример

<code>e2m.in</code>	<code>e2m.out</code>
3 3 1 2 2 3 1 3	0 1 1 1 0 1 1 1 0

Задача D. От матрицы смежности к списку ребер

Имя входного файла: `m2e.in`
Имя выходного файла: `m2e.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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

Формат входного файла

Входной файл содержит число N ($1 \leq N \leq 100$) — число вершин в графе, и затем N строк по N чисел, каждое из которых равно 0 или 1 — его матрицу смежности.

Формат выходного файла

Выведите в выходной файл список ребер заданного графа. Ребра можно выводить в произвольном порядке.

Пример

m2e.in	m2e.out
3	1 2
0 1 1	2 3
1 0 1	1 3
1 1 0	

Задача Е. Компоненты связности

Имя входного файла: **matrix.in**
Имя выходного файла: **matrix.out**
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности.

Формат входного файла

В первой строке входного файла содержится одно натуральное число N ($N \leq 100$) — количество вершин в графе. Далее в N строках по N чисел — матрица смежности графа: в i -ой строке на j -ом месте стоит 1, если вершины i и j соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

Формат выходного файла

Вывести одно целое число — искомое количество компонент связности графа.

Пример

matrix.in	matrix.out
6	3
0 1 1 0 0 0	
1 0 1 0 0 0	
1 1 0 0 0 0	
0 0 0 0 1 0	
0 0 0 1 0 0	
0 0 0 0 0 0	

Задача F. Лесопосадки

Имя входного файла: **tree.in**
Имя выходного файла: **tree.out**
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дан неориентированный невзвешенный граф. Необходимо определить, является ли он деревом.

Формат входного файла

В первой строке входного файла содержится одно натуральное число N ($N \leq 100$) — количество вершин в графе. Далее в N строках по N чисел — матрица смежности графа: в i -ой строке на j -ом месте стоит 1, если вершины i и j соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

Формат выходного файла

Вывести YES, если граф является деревом, NO иначе.

Пример

tree.in	tree.out
6 0 1 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	NO
3 0 1 0 1 0 1 0 1 0	YES

Задача G. Поиск цикла

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

Дан ориентированный невзвешенный граф. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

Формат входного файла

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N \leq 100000$, $M \leq 100000$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходного файла

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

Пример

cycle.in	cycle.out
2 2 1 2 2 1	YES 2 1
2 2 1 2 1 2	NO