

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	3
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	

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	3 3 2 2 5 2 3
5 1	
3 2	
7 1	
5 2	
7 4	
6 5	
6 4	
7 5	
2 1	
5 3	

Задача С. Цветной дождь

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

В Банановой республике очень много холмов, соединенных мостами. На химическом заводе произошла авария, в результате чего испарилось экспериментальное удобрение «зюван». На следующий день выпал цветной дождь, причем он прошел только над холмами, в некоторых местах падали красные капли, в некоторых — синие, а в остальных — зеленые, в результате чего холмы стали соответствующего цвета. Президенту Банановой республики это понравилось, но ему захотелось покрасить мосты между вершинами холмов так, чтобы мосты были покрашены в цвет холмов, которые они соединяют. К сожалению, если холмы разного цвета, то покрасить мост таким образом не удастся. Посчитать количество таких «плохих» мостов.

Формат входного файла

Во входном файле вводится N ($1 \leq N \leq 100$) — число холмов. Далее идет матрица размером $N \times N$, описывающая наличие мостов между холмами: число 1 в позиции (i, j) обозначает, что мост между холмами i и j есть, 0 — что моста нет. Матрица симметрична относительно главной диагонали, на главной диагонали стоят нули. В последней строке записано N чисел, обозначающих цвет холмов: 1 — красный; 2 — синий; 3 — зеленый.

Формат выходного файла

В выходной файл выведите количество «плохих» мостов.

Пример

rain.in	rain.out
7	4
0 1 0 0 0 1 1	
1 0 1 0 0 0 0	
0 1 0 0 1 1 0	
0 0 0 0 0 0 0	
0 0 1 0 0 1 0	
1 0 1 0 1 0 0	
1 0 0 0 0 0 0	
1 1 1 1 1 3 3	

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 E. 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

e2m.in	e2m.out
3 3	0 1 1
1 2	1 0 1
2 3	1 1 0
1 3	

Задача F. 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 G. 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

<code>tree.in</code>	<code>tree.out</code>
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 H. Операции на графе

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

В этой задаче необходимо организовать неориентированный граф, на котором поддерживаются следующие операции:

1. `AddEdge(u, v)` — добавить в граф ребро между вершинами (u, v)
2. `Vertex(u)` — вывести список вершин, смежных с вершиной u

Петель и кратных ребер в графе нет.

Формат входного файла

В первой строке входного файла содержится целое число N ($1 \leq N \leq 10^5$) — количество вершин в графе. В следующей строке находится целое число K ($0 \leq K \leq 10^6$) — число операций, затем идет описание операций — каждое в своей строке. Операции имеют следующий формат: “1 <число> <число>” или “2 <число>”, обозначающие соответственно операции `AddEdge(u, v)` и `Vertex(u)`.

Гарантируется, что суммарное количество чисел, которое будет необходимо вывести при выполнении всех операций `Vertex` не превосходит $2 \cdot 10^5$.

Формат выходного файла

В выходной файл для каждой команды `Vertex` необходимо на отдельной строке вывести список смежных вершин указанной вершины. Вершины списка смежности можно выводить в произвольном порядке.

Пример

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