

Задача А. 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 В. 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 С. 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 D. 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 E. 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 F. 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	
3	1 4
	1 2
2	

Задача G. 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	