

Задача А. Банкет

Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

На банкет были приглашены N Очень Важных Персон (ОВП). Были поставлены 2 стола. Столы достаточно большие, чтобы все посетители банкета могли сесть за любой из них. Проблема заключается в том, что некоторые ОВП не ладят друг с другом и не могут сидеть за одним столом. Вас попросили определить, возможно ли всех ОВП рассадить за двумя столами.

Формат входных данных

В первой строке входного файла дано два числа: N и M ($1 \leq N \leq 100$, $1 \leq M \leq 100$), где N — количество ОВП, а M — количество пар ОВП, которые не могут сидеть за одним столом. В следующих M строках записано по 2 числа — пары ОВП, которые не могут сидеть за одним столом.

Формат выходных данных

Если способ рассадить ОВП существует, то в выходной файл выведите YES в первой строке и номера ОВП, которых необходимо посадить за первый стол, во второй строке. В противном случае в первой и единственной строке выведите NO.

Пример

Входные данные	Выходные данные
3 2 1 2 1 3	YES 2 3

Задача В. Построение

Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Группа солдат-новобранцев прибыла в армейскую часть N666. После знакомства с прапорщиком стало очевидно, что от работ на кухне по очистке картофеля спасти солдат может только чудо.

Прапорщик, будучи не в состоянии запомнить фамилии, пронумеровал новобранцев от 1 до N . После этого он велел им построиться по росту (начиная с самого высокого). С этой несложной задачей могут справиться даже совсем необученные новобранцы, да вот беда, прапорщик уверил себя, что знает про некоторых солдат, кто из них кого выше, и это далеко не всегда соответствует истине.

После трех дней обучения новобранцам удалось выяснить, что знает (а точнее, думает, что знает) прапорщик. Помогите им, используя эти знания, построиться так, чтобы товарищ прапорщик остался доволен.

Формат входных данных

Во входном файле сначала идут числа N и M ($1 \leq N \leq 100$, $1 \leq M \leq 5000$) — количество солдат в роте и количество пар солдат, про которых прапорщик знает, кто из них выше. Далее идут эти пары чисел A и B по одной на строке ($1 \leq A \leq N$, $1 \leq B \leq N$), что означает, что, по мнению прапорщика, солдат A выше, чем B .

Формат выходных данных

В первой строке выведите "Yes" (если можно построиться так, чтобы прапорщик остался доволен) или "No" (если нет). После ответа "Yes" на следующей строке выведите N чисел разделенных пробелами, — одно из возможных построений.

Пример

Входные данные	Выходные данные
5 4 1 3 1 4 4 3 5 2	Yes 5 2 1 4 3

Задача С. Четный граф

Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Неориентированный граф называется четно-нечетным, если найдутся две его вершины, между которыми существуют пути как из четного, так и из нечетного числа ребер. Напишите программу, которая:

- определяет, является ли заданный граф четно-нечетным;
- В случае отрицательного ответа на пункт а) находит максимальное подмножество X вершин графа такое, что для любых двух вершин i и j из X выполняется следующее условие: все пути между i и j состоят из четного числа ребер.

Формат входных данных

Первая строка входного файла содержит число вершин графа N ($1 \leq N \leq 100$) и число ребер M ($1 \leq M \leq 1000$), а каждая последующая – пару чисел (i, j) , означающих, что в графе присутствует ребро, соединяющее вершины с номерами i и j .

Формат выходных данных

Первая строка выходного файла должна содержать ответ на пункт а) в форме YES/NO. В случае отрицательного ответа на пункт а) вторая строка должна содержать количество вершин в множестве X , а третья – номера вершин из этого множества в порядке возрастания, записанные через пробел. Если вариантов решений несколько, то достаточно вывести любое из них.

Примеры

Входные данные	Выходные данные
3 1 1 2	NO 2 2 3

Задача D. Водостоки

Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Карту местности условно разбили на квадраты, и посчитали среднюю высоту над уровнем моря для каждого квадрата.

Когда идет дождь, вода равномерно выпадает на все квадраты. Если один из четырех соседних с данным квадратом квадратов имеет меньшую высоту над уровнем моря, то вода с текущего квадрата стекает туда (и, если есть возможность, то дальше), если же все соседние квадраты имеют большую высоту, то вода *скапливается* в этом квадрате.

Разрешается в некоторых квадратах построить водостоки. Когда на каком-то квадрате строят водосток, то вся вода, которая раньше скапливалась в этом квадрате, будет утекать в водосток.

Если есть группа квадратов, имеющих одинаковую высоту и образующих связную область, то если хотя бы рядом с одним из этих квадратов есть квадрат, имеющий меньшую высоту, то вся вода утекает туда, если же такого квадрата нет, то вода стоит во всех этих квадратах. При этом достаточно построить водосток в любом из этих квадратов, и вся вода с них будет утекать в этот водосток.

Требуется определить, какое минимальное количество водостоков нужно построить, чтобы после дождя вся вода утекала в водостоки.

Формат входных данных

Во входном файле записаны сначала числа N и M , задающие размеры карты — натуральные числа, не превышающие 100. Далее идет N строк, по M чисел в каждой, задающих высоту квадратов карты над уровнем моря. Высота задается натуральным числом, не превышающим 10000. Считается, что квадраты, расположенные за пределами карты, имеют высоту 10001 (то есть вода никогда не утекает за пределы карты).

Формат выходных данных

В выходной файл выведите минимальное количество водостоков, которое необходимо построить.

Пример

Входные данные	Выходные данные
4 5 1 2 3 1 10 1 4 3 10 10 1 5 5 5 5 6 6 6 6 6	2

Задача Е. Кратеры на Луне

Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Пролетающие время от времени в опасной близости от нашего спутника Луны астероиды захватываются ее гравитационным полем и, будучи ничем не задерживаемы, врезаются с огромной скоростью в лунную поверхность, оставляя в память о себе порядочных размеров кратеры приблизительно круглой формы.

Увлекающийся астрономией профессор З. В. Ездочетов занялся изучением современной карты участка лунной поверхности. Он решил найти на ней максимально длинную цепочку вложенных друг в друга кратеров. Зная о Ваших недюжинных способностях в области построения алгоритмов, за помощью в решении этой непростой задачи он обратился к Вам.

Формат входных данных

Первая строка входного файла содержит целое число N – количество кратеров, отмеченных на карте ($1 \leq N \leq 500$). Следующие N строк содержат описания кратеров с номерами от 1 до N . Описание каждого кратера занимает отдельную строку и состоит из трех целых чисел, принадлежащих диапазону от -32768 до 32767 включительно, и разделенных пробелами. Первые два числа представляют собой декартовы координаты его центра, а третье – радиус. Все кратеры различны.

Формат выходных данных

Первая строка выходного файла должна содержать длину искомой цепочки кратеров, вторая – номера кратеров из этой цепочки, начиная с меньшего кратера и кончая самым большим. Номера кратеров должны быть разделены пробелами. Если существует несколько длиннейших цепочек, следует вывести любую из них.

Пример

Входные данные	Выходные данные
4 0 0 30 -15 15 20 15 10 5 10 10 10	3 3 4 1

Задача F. Почтальон

Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

В городе есть N площадей, соединенных улицами. При этом количество улиц не превышает 100000 и существует не более трех площадей, на которые выходит нечетное количество улиц. Для каждой улицы известна ее длина. По каждой улице разрешено движение в обе стороны. В городе есть хотя бы одна улица. От каждой площади до любой другой можно дойти пол улицам.

Почтальону требуется пройти хотя бы один раз по каждой улице. Почтальон хочет, чтобы длина его пути была минимальна. Он может начать движение на любой площади и закончить также на любой (в том числе и на начальной).

Помогите почтальону составить такой маршрут.

Формат входных данных

Сначала записано число N — количество площадей в городе ($2 \leq N \leq 1000$). Далее следуют N строк, задающих улицы. В i -ой из этих строк находится число m_i — количество улиц, выходящих из площади m_i . Далее следуют m_i пар натуральных чисел: в j -ой паре первое число — номер площади, в которую идет j -ая улица с i -ой площади, а второе число — длина этой улицы.

Между двумя площадями может быть несколько улиц, но не может быть улицы с площади на нее саму.

Все числа во входном файле не превосходят 100000.

Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число — количество улиц в искомом маршруте, а во вторую — номера площадей в порядке их посещения.

Если решения нет, выведите в выходной файл одно число -1 .

Если решений несколько, выведите любое.

Примеры

Входные данные	Выходные данные
<pre> 4 2 2 1 2 2 4 1 2 4 4 3 5 1 1 2 2 5 4 8 2 3 8 2 4 </pre>	<pre> 5 1 2 3 4 2 1 </pre>