

Задача А. Волновой обход графа

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Пусть *расстояние* от вершины u до вершины v — это минимальное количество рёбер в пути между u и v ; так, расстояние между u и u — 0, а расстояние между любыми двумя различными соседними вершинами — 1.

Волновым обходом графа из вершины v назовём такую последовательность вершин u_1, u_2, \dots, u_r , что:

- $u_1 = v$,
- Каждая вершина графа, достижимая из v , встречается в ней хотя бы один раз, и
- Каждая следующая вершина последовательности удалена от вершины v не меньше, чем предыдущая.

Задан связный неориентированный граф и его вершина v . Выведите любой волновой обход этого графа.

Формат входных данных

В первой строке заданы числа N , M и v через пробел — количество вершин и рёбер в графе и начальная вершина обхода ($1 \leq N \leq 100$, $0 \leq M \leq 10\,000$, $1 \leq v \leq N$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leq u_i, v_i \leq N$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i .

Формат выходных данных

В первой строке выведите число r — количество вершин в найденном волновом обходе ($1 \leq r \leq 10\,000$; гарантируется, что обход, удовлетворяющий этим ограничениям, существует). Во второй строке выведите сами числа u_1, u_2, \dots, u_r через пробел.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 1 2 2 3	3 1 2 3
4 4 1 1 2 2 3 3 4 4 1	4 1 2 4 3

Задача В. Пошаговый обход графа

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Пошаговым обходом графа из вершины v назовём такую последовательность вершин u_1, u_2, \dots, u_r , что:

- $u_1 = u_r = v$,
- Каждая вершина графа, достижимая из v , встречается в ней хотя бы один раз, и
- Между любыми двумя соседними вершинами последовательности в графе существует ребро.

Задан связный неориентированный граф и его вершина v . Выведите любой пошаговый обход этого графа.

Формат входных данных

В первой строке заданы числа N , M и v через пробел — количество вершин и рёбер в графе и начальная вершина обхода ($1 \leq N \leq 100$, $0 \leq M \leq 10\,000$, $1 \leq v \leq N$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leq u_i, v_i \leq N$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i .

Формат выходных данных

В первой строке выведите число r — количество вершин в найденном пошаговом обходе ($1 \leq r \leq 10\,000$; гарантируется, что обход, удовлетворяющий этим ограничениям, существует). Во второй строке выведите сами числа u_1, u_2, \dots, u_r через пробел.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 1 2 2 3	5 1 2 3 2 1
4 4 1 1 2 2 3 3 4 4 1	5 1 2 3 4 1

Задача С. Связность

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В этой задаче требуется проверить, что граф является *связным*, то есть что из любой вершины можно по рёбрам этого графа попасть в любую другую.

Формат входных данных

В первой строке заданы числа N и M через пробел — количество вершин и рёбер в графе, соответственно ($1 \leq N \leq 100$, $0 \leq M \leq 10\,000$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leq u_i, v_i \leq N$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i .

Формат выходных данных

Выведите «YES», если граф является связным, и «NO» в противном случае.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 2 3 2	YES
3 1 1 3	NO

Задача D. Дерево

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Проверьте, является ли он деревом.

Формат входных данных

В первой строке входных данных заданы через пробел два целых числа n и m — количество вершин и рёбер в графе, соответственно ($1 \leq n \leq 100$). В следующих m строках заданы рёбра; i -я из этих строк содержит два целых числа u_i и v_i через пробел — номера концов i -го ребра ($1 \leq u_i, v_i \leq n$). Граф не содержит петель и кратных рёбер.

Формат выходных данных

В первой строке выведите «YES», если граф является деревом, и «NO» в противном случае.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 2 1 3	YES
3 3 1 2 2 3 3 1	NO

Задача Е. Расстояние от корня

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В заданном корневом дереве найдите вершины, максимально удалённые от корня. Расстоянием между вершинами считается количество рёбер в пути.

Формат входных данных

В первой строке задано n — количество вершин в дереве ($1 \leq n \leq 100$). В следующих $n - 1$ строках заданы вершины, являющиеся предками вершин $2, 3, \dots, n$. Вершина 1 является корнем дерева.

Формат выходных данных

В первой строке выведите максимальное расстояние от корня до остальных вершин дерева. Во второй строке выведите, сколько вершин дерева находятся от корня на таком расстоянии. В третьей строке выведите номера этих вершин через пробел в порядке возрастания.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	1
1	2
1	2 3
3	2
1	1
2	3

Задача F. (p, q) -лошадь

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

(p, q) -лошадь — это обобщение обычного шахматного коня. (p, q) -лошадь своим ходом перемещается на p клеток в одном направлении и на q — в другом (перпендикулярном). Например, $(3, 4)$ -лошадь может переместиться с клетки $(5, 6)$ на клетки $(1, 3)$, $(2, 2)$, $(2, 10)$, $(1, 9)$, $(8, 10)$, $(9, 9)$, $(8, 2)$ и $(9, 3)$. Очевидно, что обычный шахматный конь — это $(2, 1)$ -лошадь.

Ваша задача — определить минимальное число ходов, которое требуется (p, q) -лошади, чтобы добраться от одной клетки шахматной доски $M \times N$ до другой. За пределы доски выходить запрещается.

Формат входных данных

Единственная строка входных данных содержит восемь целых чисел $M, N, p, q, x_1, y_1, x_2, y_2$ ($1 \leq x_1, x_2 \leq M \leq 100, 1 \leq y_1, y_2 \leq N \leq 100, 0 \leq p \leq 100, 0 \leq q \leq 100$).

Формат выходных данных

В первой строке выведите целое число k — минимальное число ходов, которое требуется (p, q) -лошади, чтобы добраться из клетки (x_1, y_1) в клетку (x_2, y_2) . Далее должны следовать $k + 1$ строк, в которых должны быть записаны последовательные положения (p, q) -лошади на этом пути.

Если (p, q) -лошадь не может добраться из (x_1, y_1) в (x_2, y_2) , выведите единственное число -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 1 1 1 1 3 3	2 1 1 2 2 3 3
2 2 1 1 1 1 1 2	-1

Задача G. Топологическая сортировка

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать: найти такую перестановку вершин, чтобы для каждого ребра графа его начало встречалось в этой перестановке раньше, чем конец.

Формат входных данных

В первой строке даны два натуральных числа N и M — количество вершин и рёбер в графе соответственно ($1 \leq N \leq 100\,000$, $0 \leq M \leq 100\,000$). Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой: номерами начальной и конечной вершин соответственно.

Формат выходных данных

Выведите любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, выведите одно число -1 . Если возможных перестановок несколько, выведите любую из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5
3 3 1 2 2 3 3 1	-1

Задача Н. Кодовый замок

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Петя опоздал на тренировку по программированию! Поскольку тренировка проходит в воскресенье, главный вход в учебный корпус, где она проходит, оказался закрыт, а вахтёр ушёл куда-то по своим делам. К счастью, есть другой способ проникнуть в здание — открыть снаружи боковую дверь, на которой установлен кодовый замок.

На пульте замка есть d кнопок с цифрами $0, 1, \dots, d - 1$. Известно, что код, открывающий замок, состоит из k цифр. Замок открывается, если последние k нажатий кнопок образуют код.

Поскольку Петя не имеет понятия, какой код открывает замок, ему придётся перебрать все возможные коды из k цифр. Но, чтобы как можно скорее попасть на тренировку, нужно минимизировать количество нажатий на кнопки. Помогите Пете придумать такую последовательность нажатий на кнопки, при которой все возможные коды были бы проверены, а количество нажатий при этом оказалось бы минимально возможным.

Формат входных данных

В первой строке записаны через пробел два целых числа d и k — количество кнопок на пульте и размер кода, соответственно ($2 \leq d \leq 10$, $1 \leq k \leq 20$).

Формат выходных данных

В первой строке выведите искомую последовательность. Если последовательностей минимальной длины, перебирающих все возможные коды, несколько, можно выводить любую из них. Гарантируется, что d и k таковы, что минимальная длина последовательности не превосходит одного мегабайта.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3	0001011100

Пояснение к примеру

Последовательность в примере перебирает все коды длины 3 в следующем порядке: 000, 001, 010, 101, 011, 111, 110, 100.

Задача I. Мосты

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Требуется найти все мосты в нём.

Формат входных данных

Первая строка содержит два натуральных числа n и m — количество вершин и рёбер графа соответственно ($n \leq 20\,000$, $m \leq 200\,000$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i и e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Формат выходных данных

Первая строка должна содержать одно натуральное число b — количество мостов в заданном графе. В следующей строке выведите b целых чисел — номера рёбер, которые являются мостами, в возрастающем порядке. Рёбра нумеруются с единицы в том порядке, в котором они заданы во входных данных.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
6 7	1
1 2	3
2 3	
3 4	
1 3	
4 5	
4 6	
5 6	

Задача J. Точки сочленения

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам задан неориентированный связный граф с N вершинами и M рёбрами ($1 \leq N \leq 20\,000$, $1 \leq M \leq 200\,000$). В графе отсутствуют петли и кратные рёбра.

Найдите все точки сочленения в заданном графе.

Формат входных данных

Граф задан следующим образом: первая строка содержит числа N и M . Каждая из следующих M строк содержит описание ребра: два целых числа из диапазона от 1 до N — номера концов ребра.

Формат выходных данных

В первой строке выведите число C — количество точек сочленения в заданном графе. В следующей строке выведите C целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

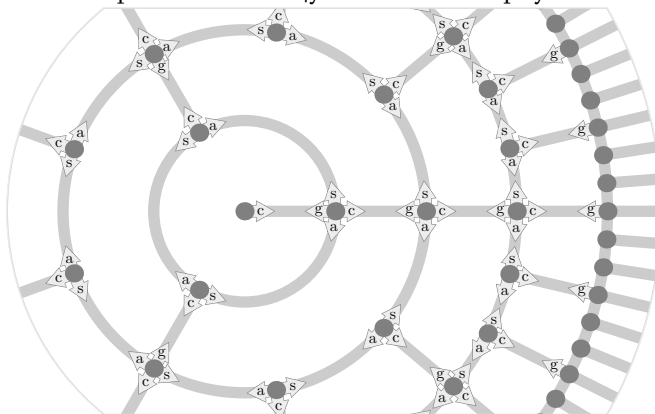
Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 6	1
1 2	3
2 3	
1 3	
3 4	
3 5	
4 5	

Задача К. Галактический Центр

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Эрнест и Селестина прибыли в Галактический Центр, чтобы зарегистрировать свой домик на краю Галактики. Центр — это огромная система залов и коридоров. Эрнест и Селестина очутились в центральном зале и посмотрели на висящую на стене карту ближайших окрестностей.



Залы Центра делятся на уровни: на уровне 0 лишь один зал — центральный, на уровне 1 три зала, на уровне 2 — девять залов, на третьем уровне целых 27 залов, и так далее. Галактический центр поистине огромен, так что количество уровней можно считать бесконечным. Залы каждого уровня, кроме нулевого, соединены коридорами в кольцо. Кроме того, каждый зал соединён коридором с залом следующего уровня. Соответственно, каждый третий зал соединяется с залом предыдущего уровня. Коридоры образуют регулярную структуру, как показано на карте выше. По всем коридорам можно перемещаться в обе стороны. Удивительно, но перемещение по любому коридору в любую сторону занимает одно и то же фиксированное время.

Маршруты перемещений в Центре записываются как строки из маленьких английских букв. Каждая буква означает движение по коридору до соседнего зала. Буквы соответствуют следующим направлениям на карте: «s» (спин) — против часовой стрелки, «a» (антиспин) — по часовой стрелке, «c» (центробежное) — от центра, «g» (гравитационное) — к центру. Адресом зала считается любой маршрут, начинающийся в центральном зале и ведущий в этот зал.

Эрнесту и Селестине нужно сначала поставить подпись в зале по адресу s , а затем — печать в зале по адресу t . Они уже справились с первой половиной этой задачи, и теперь находятся в зале по адресу s . Помогите им найти кратчайший маршрут в зал по адресу t . Длиной маршрута считается количество букв в нём.

Формат входных данных

В первой строке задан адрес s , а во второй — адрес t . Каждый из них содержит от 1 до 35 букв. Гарантируется, что оба адреса являются корректными маршрутами из центрального зала, а также что холлы по этим двум адресам различны.

Формат выходных данных

Выведите кратчайший маршрут из зала по адресу s в зал по адресу t . Если возможных ответов несколько, выведите любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
ccc csss	gg
cccscs cccacs	aaaa

Задача L. Флойд

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный взвешенный граф. В нём вам необходимо найти пару вершин, кратчайшее расстояние от одной из которых до другой максимально среди всех пар вершин.

Формат входных данных

В первой строке записано целое число N — количество вершин графа ($1 \leq N \leq 100$). В каждой из следующих N строк задано по N чисел — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое неотрицательное число — присутствие ребра данного веса.

На главной диагонали матрицы всегда стоят нули. Все числа во входных данных не превосходят 100. Гарантируется, что хотя бы одно ребро в графе присутствует.

Формат выходных данных

Вывести искомое максимальное кратчайшее расстояние.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 0 5 9 -1 -1 0 2 8 -1 -1 0 7 4 -1 -1 0	16

Задача М. Сумма расстояний

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан связный граф. Требуется найти сумму расстояний между всеми парами вершин.

Формат входных данных

Первая строка содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leq n \leq 1000$, $0 \leq m \leq 10\,000$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i , e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Гарантируется, что граф связан.

Формат выходных данных

Первая строка должна содержать одно натуральное число — сумму попарных расстояний между вершинами.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 5 1 2 2 3 3 4 5 3 1 5	16

Задача N. Порядок циклов

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф из n вершин, заданный матрицей смежности a ($a[u][u] = \text{True}$; $a[u][v] = a[v][u]$; $a[u][v] = \text{True}$ тогда и только тогда, когда есть ребро между вершинами u и v). На нём запускают следующий алгоритм:

```
for x := 1 to n do
  for y := 1 to n do
    for z := 1 to n do
      if a[i][k] and a[k][j] then
        a[i][j] := True;
```

Перед запуском буквы x , y и z заменяют буквами i , j и k в некотором порядке. Утверждается, что после работы этого алгоритма $a[u][v] = \text{True}$ тогда и только тогда, когда в исходном графе существует путь между вершинами u и v . Выясните, верно ли это, и если нет, приведите пример исходного графа, на котором это неверно.

Формат входных данных

В первой строке ввода записаны через пробел три буквы — «i», «j» и «k» — в некотором порядке. Первая буква подставляется в программу вместо «x», вторая — вместо «y», третья — вместо «z».

Формат выходных данных

Если искомый граф существует, в первой строке выведите через пробел целые числа n и m — количество вершин и рёбер в графе, соответственно ($1 \leq n \leq 10$, $0 \leq m \leq 45$). В следующих m строках выведите пары вершин, соединённых рёбрами, по одной паре на строке. Номера вершин в паре должны быть упорядочены по возрастанию; вершины нумеруются с единицы. Кратные рёбра и петли не допускаются.

Если же программа с заданным порядком циклов корректно работает на любом графе, вместо n и m выведите в первой строке два нуля через пробел.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
k i j	0 0

Задача О. Цикл отрицательного веса

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный граф. Определите, есть ли в нём цикл отрицательного веса, и если да, то выведите его.

Формат входных данных

В первой строке задано число N — количество вершин графа ($1 \leq N \leq 100$). В следующих N строках находится по N чисел — матрица смежности графа. Все веса рёбер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

Формат выходных данных

В первой строке выведите «YES», если цикл существует или «NO» в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле и в третьей строке — вершины, входящие в этот цикл, в порядке обхода.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	YES
0 -1	2
-1 0	1 2

Задача Р. Лабиринт знаний

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Участникам тренировки подарили билеты на аттракцион «Лабиринт знаний». Лабиринт представляет собой N комнат, пронумерованных целыми числами от 1 до N , между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определённую величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход — в комнате N . Каждый участник сборов проходит лабиринт ровно один раз и набирает некоторое количество знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача — показать наилучший результат.

Формат входных данных

Первая строка содержит целые числа N и M — количество комнат и дверей ($1 \leq N \leq 2000$, $0 \leq M \leq 10\,000$). В каждой из следующих M строк содержится описание двери — номер комнаты, из которой она ведёт, и номер комнаты, в которую она ведёт (через дверь в лабиринте можно ходить только в одну сторону), а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10 000). Двери могут вести из комнаты в неё саму, между двумя комнатами может быть более одной двери.

Формат выходных данных

Выведите «:»», если можно пройти лабиринт и получить неограниченно большой запас знаний, «:(», если лабиринт пройти нельзя, и максимальное количество набранных знаний в остальных случаях.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 2 1 2 5 1 2 -5	5

Задача Q. Синхронизация

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Учёные взялись за какую-то большую и сложную проблему. Для её решения написали распределённое приложение. Процессы приложения во время решения задачи должны обмениваться сообщениями. Каждый учёный запустил один или несколько процессов приложения на своём компьютере, и спустя некоторое время решение проблемы было найдено.

После того, как решение было получено, возник вопрос о том, как происходил обмен данными между компьютерами. Но вот незадача — времена на компьютерах не были синхронизированы, и осталась только информация о времени запуска приложения, времена вызовов функций пересылки и приёма данных и их продолжительность.

Ваша задача — найти, на сколько отличаются времена на компьютерах, на которых считалась задача. Но из-за того, что время отправления сигнала регистрировалось в момент окончания отправления, а время получения сигнала регистрировалось в момент начала приёма сигнала, может сложиться такая ситуация, что определение возможной разности времён на компьютерах невозможно.

Формат входных данных

В первой строке вводятся через пробел два числа N и M ($1 \leq N \leq 100$, $0 \leq M \leq 10\,000$), где N — количество компьютеров, а M — количество пересланных пакетов данных. В следующих M строках задаются параметры пакетов данных: в каждой строке по три числа, записанных через пробел, i , j и Δt ($1 \leq i \leq N$, $1 \leq j \leq N$, $-10\,000 \leq \Delta t \leq 10\,000$). Здесь i — номер компьютера, получившего пакет данных, j — номер компьютера, пославшего пакет данных, а Δt — разница между временем начала получения i -м компьютером и временем окончания отправки j -м компьютером пакета данных. В Δt входят относительные времена, которые измерялись на i -м и j -м компьютерах соответственно. Поэтому разность реальных времён начала получения и окончания отправки данных могла быть отлична от Δt , если времена на компьютерах отличались.

Формат выходных данных

Выведите N чисел, по одному в каждой строке. В i -й строке должна содержаться возможная разность между временем i -го компьютера и сервера (сервер — компьютер с номером 1). При этом рассчитанные времена должны удовлетворять тому условию, что момент начала получения был не раньше момента окончания отправки каждого из пакетов данных. Если возможных наборов времён несколько, то нужно выдать любой из них. Если возможных наборов времён нет, то необходимо выдать строку «NO SOLUTION» (без кавычек).

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 6	0
1 2 5	2
2 1 5	1
1 3 1	
3 1 1	
2 3 1	
3 2 1	

Задача R. Дейкстра

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный взвешенный граф. Найдите кратчайшее расстояние от одной заданной вершины до другой.

Формат входных данных

В первой строке даны три числа: N , S и F ($1 \leq N \leq 1000$, $1 \leq S, F \leq N$), где N — количество вершин графа, S — начальная вершина, а F — конечная. В следующих N строках задано по N чисел — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое неотрицательное число — присутствие ребра данного веса. На главной диагонали матрицы всегда стоят нули. Веса всех рёбер целые и не превосходят 10 000.

Формат выходных данных

Вывести искомое расстояние или -1 , если пути не существует.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 2 0 -1 2 3 0 -1 -1 4 0	6

Задача S. Соединение точек

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны N точек на плоскости. Требуется провести отрезки между некоторыми парами точек таким образом, чтобы, во-первых, из любой данной точки в любую можно было пройти по этим отрезкам, а во-вторых, суммарная длина проведённых отрезков была минимальна.

Формат входных данных

В первой строке входных данных задано число N — количество точек ($1 \leq N \leq 200$). Следующие N строк содержат по два числа X_i Y_i каждая через пробел — координаты i -й точки ($-1000 \leq X_i, Y_i \leq 1000$). Никакие две данные точки не совпадают, никакие три не лежат на одной прямой. Все числа во входных данных целые.

Формат выходных данных

В первой строке выведите L — суммарную длину проведённых отрезков с точностью не менее шести знаков после десятичной точки. Во второй строке выведите K — их количество. В следующих K строках выведите по два числа A_j B_j через пробел в каждой — номера точек, соединённых j -м отрезком ($1 \leq A_j, B_j \leq N$, $A_j \neq B_j$). Точки нумеруются с единицы в том порядке, в котором они даны во входных данных. Если ответов с минимальным L несколько, разрешается выводить любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 0 0 0 1 1 0 1 1	3 3 1 2 2 4 4 3
5 0 0 0 2 1 1 3 0 3 2	7.064495 4 3 1 3 2 3 4 4 5

Задача Т. Подграф

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Требуется удалить из него некоторые рёбра так, чтобы получился связный подграф, содержащий все вершины исходного графа, в котором суммарная длина всех рёбер минимальна.

Формат входных данных

В первой строке заданы два целых числа n и m через пробел — количество вершин и рёбер в исходном графе, соответственно ($1 \leq n \leq 100$, $m \geq 0$). Следующие m строк описывают рёбра; i -я из них содержит три целых числа u_i , v_i и w_i — номера концов ребра и его длину ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $|w_i| \leq 10\,000$). Гарантируется, что между каждой парой вершин есть не более одного ребра.

Формат выходных данных

Если невозможно удалить рёбра так, чтобы получился связный подграф, выведите «Impossible» (без кавычек). В противном случае выведите одно число — минимальную сумму длин рёбер подграфа.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 1 2 3 2 3 4 3 1 5	7