

Задача А. Любители Кошек

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 мегабайт

В университетском клубе любителей кошек зарегистрировано n членов. Естественно, что некоторые из членов клуба знакомы друг с другом. Нужно сосчитать, сколькими способами можно выбрать из них троих, которые могли бы свободно общаться (то есть, любые два из которых знакомы между собой).

Формат входных данных

В первой строке ввода заданы числа n и m ($1 \leq n \leq 1000$, $1 \leq m \leq 30\,000$), где m обозначает общее число знакомств. В последующих m строках идут пары чисел a_i b_i , обозначающие, что a_i знаком с b_i . Информация об одном знакомстве может быть записана несколько раз, причём даже в разном порядке (как (x, y) , так и (y, x)).

Формат выходных данных

Выведите одно число — количество способов выбрать троих попарно знакомых друг с другом людей из клуба.

Пример

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

Задача В. Связность

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 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 С. Дерево

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 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 D. Расстояние от корня

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 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, ..., 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	1 2 2 3
3 1 2	2 1 3

Задача Е. Волновой обход графа

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 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 G. Лего

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 мегабайт

Рассмотрим трёхмерную конструкцию из блоков, аналогичных блокам известного конструктора «Лего». Конструкция состоит из нескольких прямоугольных слоёв одинакового размера, расположенных один над другим. Каждый слой состоит из блоков. Каждый блок — это плоская фигура из кубиков, связанная по стороне. Соседние слои скреплены между собой таким образом, что нижний кубик в каждом столбике скреплен с верхним. Различные же блоки в одном слое никак не скреплены между собой.

Два блока A и B считаются связанными, если существует такая цепочка блоков $A = C_1, C_2, C_3, \dots, C_{n-1}, C_n = B$, что каждые два соседних блока C_i и C_{i+1} в ней скреплены. Очевидно, всю конструкцию можно поделить на компоненты связности. Найдите их количество.

Формат входных данных

В первой строке ввода записаны через пробел три числа x, y и z — длина, ширина и высота конструкции, соответственно ($1 \leq x, y, z \leq 100$). Далее размещены z блоков по y строк, описывающие конструкцию. В каждой из этих строк ровно x символов. Перед каждым блоком расположена дополнительно одна пустая строка. Каждый кубик описывается одной заглавной буквой английского алфавита. Два кубика в одном слое принадлежат одному блоку, если они соседние по стороне и обозначены одной и той же буквой.

Формат выходных данных

Выведите одно число — количество компонент связности в заданной конструкции.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 2 2 AA AA AB CD	1
3 3 2 AAB BCB BAA CAA CEB DDB	2
4 3 1 AABV ACCB AABV	3

Пояснения к примерам

В первом примере четыре отдельных кубика второго слоя крепятся к одному блоку, занимающему весь первый слой. В конструкции одна компонента связности.

Во втором примере блоки из двух кубиков образуют связный «колодец». Два блока в центре связаны друг с другом, но не связаны с «колодцем». Количество компонент связности в этой конструкции равно двум.

В третьем примере в конструкции всего один слой, состоящий из трёх блоков. Отметим, что блоки не обязательно имеют прямоугольную форму. В этой конструкции три компоненты связности.

Задача Н. (p, q) -лошадь

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 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 I. Пошаговый обход графа

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 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 J. Топологическая сортировка

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 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 K. Покраска в два цвета

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 512 мегабайт

Дан неориентированный граф из n вершин и m рёбер. Нужно покрасить вершины графа в два цвета таким образом, чтобы все рёбра соединяли вершины разных цветов, или выяснить, что это невозможно.

Формат входных данных

В первой строке ввода заданы через пробел два целых числа: число вершин n и число рёбер m ($1 \leq n \leq 100$, $1 \leq m \leq 5000$). Далее следует m строк, каждая из которых содержит по два целых числа в диапазоне от 1 до n — номера вершин, которые соединяет ребро.

Формат выходных данных

Если требуемой покраски не существует, выведите -1 . В противном случае выведите через пробел n целых чисел от 1 до 2 — цвета вершин. Если решений несколько, выведите любое из них.

Примеры

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

Задача L. Мосты

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 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 ($1 \leq b_i, e_i \leq n$). Гарантируется, что в графе нет петель и кратных рёбер.

Формат выходных данных

Первая строка должна содержать одно целое 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 M. Точки сочленения

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 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 N. Неизбежность

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Вася живет в первой вершине связного неориентированного графа, состоящего из n вершин и m рёбер. Каждый день он ходит в школу, находящуюся в вершине с номером n . Вася старается каждый день ходить в школу новым маршрутом, однако однажды он заметил, что некоторые рёбра он проходит каждый день, независимо от того, каким маршрутом идёт. Помогите Васе найти все такие рёбра.

Формат входных данных

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

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

Формат выходных данных

Первая строка должна содержать одно целое число b — количество рёбер, которые неизбежно встречаются на пути Васи. В следующей строке выведите b целых чисел — номера этих рёбер в возрастающем порядке. Ребра нумеруются с единицы в том порядке, в котором они заданы во входных данных.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
7 8 1 2 2 3 3 1 4 3 5 4 5 6 4 6 6 7	2 4 8
4 4 1 2 2 4 1 4 2 3	0

Задача О. Мшистое Болото

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Мшистое Болото нарисовано на карте, имеющей форму прямоугольника. Карта поделена на квадраты. В некоторых квадратах растёт мох, в других — деревья, а остальные квадраты свободны.

Каждый день мох распространяется по болоту. Если в квадрате рос мох или дерево, то в нём ничего не меняется. А вот если квадрат K был свободен, то следует посчитать, сколько соседних квадратов уже были покрыты мхом в предыдущий день. Если этих квадратов хотя бы два, то K также покрывается мхом. При этом мох не распространяется за пределы карты. Соседними считаются квадраты, имеющие общую сторону или общий угол.

Рано или поздно во Мшистом Болоте наступит стабильность: карта болота в каждый следующий день будет точно такой же, как и в предыдущий. Глядя на карту болота сегодня, посчитайте, сколько дней пройдёт до наступления стабильности.

Формат входных данных

В первой строке заданы два целых числа r и c , разделённые пробелом — количество строк и столбцов на карте ($1 \leq r, c \leq 300$). В следующих r строках задана сама карта: каждая из этих строк содержит ровно c символов. Символ «*» обозначает клетку, поросшую мхом, символ «Т» — дерево, а символ «.» — свободную клетку. Каждая строка входных данных, включая последнюю, заканчивается переводом строки.

Формат выходных данных

Выведите одно целое число: сколько дней пройдёт до наступления стабильности во Мшистом Болоте.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 7* .Т*..Т. .Т.*ТТ. ...ТТ*.	8
3 3 .Т* Т*Т *Т.	0

Задача Р. Погоня

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Рассмотрим пустое плоское поле, бесконечное во все стороны. Поле разделено на клетки — одинаковые правильные шестиугольники. По этому полю перемещаются вертолёт и вездеход. Изначально они находятся в одной клетке и смотрят в сторону одной и той же соседней клетки.

Сначала движется вертолёт. Его движение описывается строкой, в которой могут быть следующие приказы:

- «f»: переместиться на соседнюю клетку, в сторону которой он смотрит.
- «l»: повернуть на 60 градусов налево.
- «r»: повернуть на 60 градусов направо.
- «w»: построить стену в клетке, тем самым заблокировав её для вездехода.

Все приказы, кроме «f», оставляют вертолёт в той же клетке, где он был. Повторная постройка стены в клетке не изменяет её состояние.

Настало время движения вездехода. Известно, что он может переместиться в клетку, где теперь находится вертолёт, ни разу не оказавшись в клетке со стеной. Найдите кратчайший маршрут по клеткам, который позволяет это сделать, и выведите его при помощи приказов «f», «l» и «r».

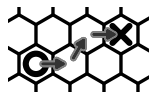
Формат входных данных

В первой строке записано целое число n — количество приказов для вертолёта ($1 \leq n \leq 5000$). Вторая строка задаёт приказы и состоит из n символов «f», «l», «r» и «w». Гарантируется, что в ней не больше 50 символов «w». Гарантируется также, что после выполнения вертолётom всех приказов требуемый маршрут для вездехода существует.

Формат выходных данных

В первой строке выведите целое число k — количество приказов для вездехода ($0 \leq k \leq 100\,000$). Во второй строке выведите сами приказы — последовательность из k символов «f», «l» и «r». Количество перемещений на соседнюю клетку, то есть букв «f», должно быть минимально возможным. Если таких ответов несколько, выведите любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>иллюстрация</i>
5 flfrf	5 flfrf	
3 fwf	8 lfrlrfrf	