Задача A. От списка ребер к матрице смежности, ориентированный вариант

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Простой ориентированный граф задан списком ребер, выведите его представление в виде матрицы смежности.

Формат входных данных

На вход программы поступают числа n ($1 \leqslant n \leqslant 100$) — количество вершин в графе и m ($1 \leqslant m \leqslant n(n-1)$) — количество ребер. Затем следует m пар чисел — ребра графа.

Формат выходных данных

Выведите матрицу смежности заданного графа.

Пример

стандартный ввод	стандартный вывод
5 3 1 3 2 3 5 2	0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0

Задача B. Компоненты связности

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности и вывести их.

Формат входных данных

Во входном файле записано два числа N и M ($0 < N \leqslant 100000, 0 \leqslant M \leqslant 100000$). В следующих M строках записаны по два числа i и $j (1 \leqslant i, j \leqslant N)$, которые означают, что вершины i и j соединены ребром.

Формат выходных данных

В первой строчке выходного файла выведите количество компонент связности. Далее выведите сами компоненты связности в следующем формате: в первой строке количество вершин в компоненте, во второй - сами вершины в отсортированном порядке.

Пример

стандартный ввод	стандартный вывод
6 4 3 1 1 2 5 4 2 3	3 3 1 2 3 2 4 5 1 6

Задача C. Долой списывание!

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Во время теста Михаил Дмитриевич заметил, что некоторые лкшата обмениваются записками. Сначала он хотел поставить им всем двойки, но в тот день Михаил Дмитриевич был добрым, а потому решил разделить лкшат на две группы: списывающих и дающих списывать, и поставить двойки только первым.

У Михаила Дмитриевича записаны все пары лкшат, обменявшихся записками. Требуется определить, сможет ли он разделить лкшат на две группы так, чтобы любой обмен записками осуществлялся от лкшонка одной группы лкшонку другой группы.

Формат входных данных

В первой строке находятся два числа N и M — количество лкшат и количество пар лкшат, обменивающихся записками ($1 \leqslant N \leqslant 100$, $0 \leqslant M \leqslant \frac{N(N-1)}{2}$). Далее в M строках расположены описания пар лкшат: два различных числа, соответствующие номерам лкшат, обменивающихся записками (нумерация лкшат идёт с 1). Каждая пара лкшат перечислена не более одного раза.

Формат выходных данных

Необходимо вывести ответ на задачу Павла Олеговича. Если возможно разделить лкшат на две группы, выведите «`YES`»; иначе выведите «`NO`».

Пример

стандартный ввод	стандартный вывод
3 2 1 2 2 3	YES

Задача D. Отбой

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

Как всем известно, в 22:45 все ЛКШата должны быть в своих домиках. Петя слишком долго играл с друзьями в настольные игры и не успел вернуться до мягкого отбоя. Теперь он задумался о том, как пройти в свой домик так, чтобы его не заметили преподаватели.

ЛКШ проводится на базе Стёпаново, на которой есть n домиков, пронумерованных от 1 до n. Домики соединяют m дорожек. Известно, что каждая дорожка соединяет ровно два различных домика, и любые два домика соединены не более, чем одной дорожкой. Передвигаться можно только по дорожкам, при этом по любой дорожке можно двигаться в любом направлении. Известно, что из любого домика можно добраться до любого другого по дорожкам.

Петя знает, что в домике t живет Денис Павлович и, если пройти мимо, то он что-то заподозрит и поднимет тревогу. Если это произойдет, то ровно на $\lfloor \frac{m}{2} \rfloor$ случайных дорожках появятся преподаватели, жаждущие поставить кому-нибудь дырку в бейджик. По этим дорожкам Петя пройти не сможет. Петя не может заранее узнать, где будут преподаватели.

Вам нужно для k пар домиков u_i и v_i определить, сможет ли Петя гарантированно добраться из домика u_i в домик v_i, не получив дырку в бейджик.

Формат входных данных

В первой строке заданы три целых числа n, m и t ($3 \leqslant n, m \leqslant 10^5$, $1 \leqslant t \leqslant n$) – число домиков на базе, число дорожек и номер домика, где живет Денис Павлович.

В следующих m строках задано по два целых числа a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$), обозначающих то, что домики a_i и b_i соединены дорожкой.

В следующей строке задано целое число k ($1 \leqslant k \leqslant 10^5$) – число запросов.

В следующих k строках задано по два целых числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$) – домики, для которых нужно определить, сможет ли Петя дойти от одного до другого. Известно, что Денис Павлович не живет ни в домике u_i, ни в домике v_i.

Формат выходных данных

Для каждого из k запросов выведите на отдельной строке `Yes`, если Петя не сможет дойти до своего домика незамеченным, иначе выведите `No`.

Пример

стандартный ввод	стандартный вывод
4 3 3 4 3 4 2 3 1 2 2 4 1 4	Yes No

Задача E. Степени вершин по списку ребер

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Неориентированный граф задан списком ребер. Найдите степени всех вершин графа.

Формат входных данных

Сначала вводятся числа n ($1 \leqslant n \leqslant 100$) — количество вершин в графе и m $\left(1 \leqslant m \leqslant \frac{n(n-1)}{2}\right)$ — количество ребер. Затем следует m пар чисел — ребра графа. Каждая вершина — число от 1 до n.

Формат выходных данных

Выведите n чисел — степени вершин графа. i-ое выведенное число должно быть равно степени вершины с номером i.

Пример

стандартный ввод	стандартный вывод
5 3 1 3 2 3 2 5	1 2 2 0 1

Задача F. Обход графа

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан неориентированный невзвешенный граф. Для него вам необходимо найти количество вершин, лежащих в одной компоненте связности с данной вершиной (считая эту вершину).

Формат входных данных

В первой строке входных данных содержатся два числа: N и S ($1 \leqslant N \leqslant 100$; $1 \leqslant S \leqslant N$), где N — количество вершин графа, а S — заданная вершина. В следующих N строках записано по N чисел — матрица смежности графа, в которой 0 означает отсутствие ребра между вершинами, а 1 — его наличие. Гарантируется, что на главной диагонали матрицы всегда стоят нули.

Формат выходных данных

Выведите одно целое число — искомое количество вершин.

Пример

стандартный ввод	стандартный вывод
3 1 0 1 1 1 0 0 1 0 0	3

Задача G. Есть ли цикл?

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан ориентированный граф. Требуется определить, есть ли в нем цикл.

Формат входных данных

В первой строке вводится число n - количество вершин и m - количество ребер. ($1 \leqslant n, m \leqslant 10^5$). Далее в m строках следует по 2 числа u, v - вершины графа, соединенные ребром.

Формат выходных данных

Выведите 0, если в заданном графе нет цикла, и 1, если он есть.

Примеры

стандартный ввод	стандартный вывод
4 4 1 2 2 3 3 4 4 1	1
3 2 1 2 1 3	0

Задача H. 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 построить его топологическую сортировку.

Формат входных данных

В первой строке входного файла даны два натуральных числа n и m ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 100\,000$) — число вершин и рёбер в графе соответственно. Далее в m строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — 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 -1.

Пример

стандартный ввод	стандартный вывод
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5

Задача I. Топологическая сортировка

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Задан ориентированный ациклический граф с n вершинами и m ребрами. Также задана перестановка вершин графа. Необходимо проверить, является ли данная перестановка топологической сортировкой.

Формат входных данных

В первой строке даны два числа n и m — количество вершин и ребер в графе соответственно ($1 \leqslant n, m \leqslant 10^5$). В следующих m строках заданы пары чисел u_i, v_i, означающие, что в графе есть ребро из вершины u_i в вершину v_i. В последней строке задана перестановка из n элементов.

Формат выходных данных

Выведите «YES» (без кавычек), если данная перестановка является топологической сортировкой и «NO» в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 3 2 3 1 3 1 2 2 1 3	NO
3 3 3 2 1 2 3 1 3 1 2	YES

Задача J. Цикл в графе

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Неориентированный граф задан списком ребер. Проверить, есть ли в нем простой цикл нечётной длины, и если да, то вывести любой такой.

Формат входных данных

В первой строке даны 2 целых числа n и m - количество вершин и ребер в графе ($1 \leqslant n \leqslant 100$, $0 \leqslant m \leqslant \frac{n(n-1)}{2}$). Далее в m строках пары чисел - номера вершин, соединенных ребром. Числа в паре различны, при чем каждая пара чисел всречается не более одного раза. Нумерация вешин начинается с 1.

Формат выходных данных

Если в графе нет цикла нечётной длины, выведите «`NO`». Если цикл есть, в первой стоке выведите «`YES`», во второй строке выведите длину цикла, а в третей строке - вершины, входящие в цикл, в том порядке, в котором они идут в цикле. Если в графе несколько подходящих циклов, выведите любой.

Примеры

стандартный ввод	стандартный вывод
5 5 1 2 2 3 4 3 4 1 4 5	NO
4 5 1 2 2 3 3 1 3 4 4 2	YES 3 3 2 1

Задача K. Свинки-копилки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У Васи есть n свинок-копилок, свинки занумерованы числами от 1 до n. Каждая копилка может быть открыта единственным соответствующим ей ключом или разбита.

Вася положил ключи в некоторые из копилок (он помнит, какой ключ лежит в какой из копилок). Теперь Вася собрался купить машину, а для этого ему нужно достать деньги из всех копилок. При этом он хочет разбить как можно меньшее количество копилок (ведь ему еще нужно копить деньги на квартиру, дачу, вертолет...). Помогите Васе определить, какое минимальное количество копилок нужно разбить.

Формат входных данных

В первой строке содержится число n — количество свинок-копилок ($1 \leqslant n \leqslant 100$). Далее идет n строк с описанием того, где лежит ключ от какой копилки: в i-й из этих строк записан номер копилки, в которой находится ключ от i-й копилки.

Формат выходных данных

Выведите единственное число: минимальное количество копилок, которые необходимо разбить.

Пример

стандартный ввод	стандартный вывод
4 2 1 2 4	2

Задача L. Бусинки

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

Маленький мальчик делает бусы. У него есть много пронумерованных бусинок. Каждая бусинка имеет уникальный номер — целое число в диапазоне от 1 до N. Он выкладывает все бусинки на полу и соединяет бусинки между собой произвольным образом так, что замкнутых контуров не образуется. Каждая из бусинок при этом оказывается соединенной с какой-либо другой бусинкой. Требуется определить, какое максимальное количество последовательно соединенных бусинок присутствует в полученной фигуре.

Формат входных данных

В первой строке записано число N ($1 \leqslant N \leqslant 5 \times 10^5$) — количество бусинок. В последующих $N - 1$ строках по два целых числа — номера, соединенных бусинок.

Формат выходных данных

Выведите одно число — искомое число бусинок

Примеры

стандартный ввод	стандартный вывод
2 1 2	2
5 2 1 2 3 2 4 2 5	3

Задача M. Авиаперелёты

Имя входного файла:	`avia.in`
Имя выходного файла:	`avia.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Главного конструктора Петю попросили разработать новую модель самолёта для компании «Air Бубундия». Оказалось, что самая сложная часть заключается в подборе оптимального размера топливного бака.

Главный картограф «Air Бубундия» Вася составил подробную карту Бубундии. На этой карте он отметил расход топлива для перелёта между каждой парой городов.

Петя хочет сделать размер бака минимально возможным, для которого самолёт сможет долететь от любого города в любой другой (возможно, с дозаправками в пути).

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leqslant n \leqslant 1\,000$) — число городов в Бубундии. Далее идут n строк по n чисел каждая. j-е число в i-й строке равно расходу топлива при перелёте из i-го города в j-й. Все числа не меньше нуля и меньше 10^9. Гарантируется, что для любого i в i-й строчке i-е число равно нулю.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — оптимальный размер бака.

Пример

avia.in	avia.out
4 0 10 12 16 11 0 8 9 10 13 0 22 13 10 17 0	10

Задача N. Доим коров

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

n коров Фермера Джона пронумерованы последовательно от 1 до n. Для доения коровы i требуется t_i единиц времени. Однако некоторых коров необходимо подоить раньше других (из-за их положения на ферме). Если корову A требуется подоить перед коровой B, ФД должен полностью закончить дойку коровы A, прежде чем начать дойку коровы B.

Для того, чтобы подоить всех своих коров как можно быстрее, ФД нанял большое количество доярок — достаточное для того, чтобы доить любое количество коров одновременно.

Определите минимальное количество времени, требуемое для дойки всех коров.

Формат входных данных

Первая строка содержит числа n и m — количество коров и количество ограничений, соответственно ($1 \leqslant n \leqslant 10\,000, 1 \leqslant m \leqslant 50\,000$). Следующие n строк, содержат t_i — времена, требуемые для дойки коров ($1 \leqslant t_i \leqslant 100\,000$). Следующие m строк содержат числа a_i и b_i — ограничения на дойку ($1 \leqslant a_i, b_i \leqslant n,\ a_i \neq b_i$).

Формат выходных данных

Выведите минимальное количество времени, требуемое чтобы подоить всех коров.

Пример

стандартный ввод	стандартный вывод
3 1 10 5 6 3 2	11

Задача O. Красно-синий граф

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вам дан граф на n вершинах с $\frac{n \cdot (n-1)}{2}$ ориентированными ребрами. Ребро между вершинами (u, v) существует, если $u < v$. Каждое ребро покрашено либо в синий, либо в красный цвет. Проверьте, существует ли такая пара вершин (v, u), что существует:

- путь $v \to u$, состоящий только из красных ребер
- путь $v \to u$, состоящий только из синих ребер

Формат входных данных

В первой строке вводится число n ($1 \leqslant n \leqslant 5000$). В следующих $n-1$ строках вводится описание графа.

i-я строка описания графа будет содержать строку из $n - i$ символов 'R' или 'B'. j-й символ обозначает цвет ребра $i \to (i + j)$.

Формат выходных данных

Если искомая пара v, u существует, выведите «YES». Иначе выведите «NO»

Примеры

стандартный ввод	стандартный вывод
3 RB R	YES
3 RR R	NO