Задача A. Топологическая сортировка (12,5 баллов)

Имя входного файла:	`topsort.in`
Имя выходного файла:	`topsort.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входного файла

В первой строке входного файла даны два целых числа N и M ($1 \leqslant N \leqslant 100\,000, 0 \leqslant M \leqslant 100\,000$) — количества 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 выходного файла

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести «-1».

Примеры

topsort.in	topsort.out
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5

Задача B. Предок (12,5 баллов)

Имя входного файла:	`ancestor.in`
Имя выходного файла:	`ancestor.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Напишите программу, которая для двух вершин дерева определяет, является ли одна из них предком другой.

Формат входного файла

Первая строка входного файла содержит натуральное число n ($1 \leqslant n \leqslant 100\,000$) — количество вершин в дереве. Во второй строке находятся n чисел, i-е из которых определяет номер непосредственного родителя вершины с номером i. Если это число равно нулю, то вершина является корнем дерева.

В третьей строке находится число m ($1 \leqslant m \leqslant 100\,000$) — количество запросов. Каждая из следующих m строк содержит два различных числа a и b ($1 \leqslant a, b \leqslant n$).

Формат выходного файла

Для каждого из m запросов выведите на отдельной строке число `1`, если вершина a является одним из предков вершины b, и `0` в противном случае.

Примеры

ancestor.in	ancestor.out
6 0 1 1 2 3 3 5 4 1 1 4 3 6 2 6 6 5	0 1 1 0 0

Задача C. Авиаперелёты (25 баллов)

Имя входного файла:	`avia.in`
Имя выходного файла:	`avia.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

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 входного файла

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 выходного файла

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

Примеры

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