

## Задача А. Кратчайший путь

Имя входного файла: mindist2.in  
Имя выходного файла: mindist2.out  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

Вам дан неориентированный граф. Найдите кратчайший путь от вершины  $a$  до вершины  $b$ .

### Формат входных данных

В первой строке входного файла идут целые числа  $n$  и  $m$  ( $1 \leq n \leq 50\,000$ ,  $1 \leq m \leq 100\,000$ ) — количества вершин и рёбер соответственно. Во второй строке идут целые числа  $a$  и  $b$  — стартовая и конечная вершины соответственно. Далее идут  $m$  строк, описывающих рёбра.

### Формат выходных данных

Если пути между  $a$  и  $b$  нет, выведите единственное число  $-1$ . Иначе выведите в первой строке число  $l$  — длину кратчайшего пути между этими двумя вершинами в рёбрах, а во второй строке выведите  $l + 1$  число — вершины этого пути.

### Примеры

mindist2.in	mindist2.out
4 5 1 4 1 3 3 2 2 4 2 1 2 3	2 1 2 4
4 4 2 3 2 1 2 4 4 3 1 3	2 2 1 3

## Задача В. TopSort

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайта

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

### Формат входных данных

В первой строке входного файла даны два натуральных числа  $N$  и  $M$  ( $1 \leq N \leq 10^5, 1 \leq M \leq 10^5$ ) — количество 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

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

### Примеры

<code>stdin</code>	<code>stdout</code>
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 С. Островные государства

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

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город  $N$  на Всебайтландское сборище программистов. Разумеется, он хочет затратить при этом как можно меньше денег и помочь ему здесь, как обычно, предлагается Вам.

**В этой задаче нельзя использовать алгоритм Дейкстры!**

### Формат входных данных

В первой строке входного файла записано два числа  $N$  и  $M$  ( $1 \leq N, M \leq 100\,000$ ) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах —  $N$  чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних  $M$  строках записаны пары  $1 \leq a, b \leq N$ ,  $a \neq b$ . Каждая пара означает наличие дороги из города  $a$  в город  $b$ . По дорогам Байтландии можно двигаться в любом направлении.

### Формат выходных данных

Если искомого пути не существует, выведите единственное слово `impossible`. В противном случае в первой строке напишите минимальную стоимость и количество посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

### Примеры

стандартный ввод	стандартный вывод
7 8	0 5
1 1 1 1 2 2 1	1 2 3 4 7
1 2	
2 5	
2 3	
5 4	
4 3	
4 7	
1 6	
6 7	

## Задача D. Флойд

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Полный ориентированный взвешенный граф задан матрицей смежности. Постройте матрицу кратчайших путей между его вершинами.

Гарантируется, что в графе нет циклов отрицательного веса.

### Формат входных данных

В первой строке вводится единственное число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках по  $N$  чисел задаётся матрица смежности графа ( $j$ -е число в  $i$ -й строке — вес ребра из вершины  $i$  в вершину  $j$ ). Все числа по модулю не превышают 100. На главной диагонали матрицы — всегда нули.

### Формат выходных данных

Выведите  $N$  строк по  $N$  чисел — матрицу расстояний между парами вершин, где  $j$ -е число в  $i$ -й строке равно весу кратчайшего пути из вершины  $i$  в  $j$ .

### Примеры

<code>stdin</code>	<code>stdout</code>
4	0 5 7 13
0 5 9 100	12 0 2 8
100 0 2 8	11 16 0 7
100 100 0 7	4 9 11 0
4 100 100 0	

## Задача Е. Предок

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

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

### Формат входных данных

Первая строка содержит количество вершин в дереве  $n$  ( $1 \leq n \leq 10^5$ ). Во второй строке находится  $n$  чисел,  $i$ -ое из которых определяет номер непосредственного родителя вершины с номером  $i$ . Если это число равно нулю, то вершина является корнем дерева.

В третьей строке находится количество запросов  $m$  ( $1 \leq m \leq 10^5$ ). Каждая из следующих  $m$  строк содержит два различных числа  $a$  и  $b$  ( $1 \leq a, b \leq n$ ).

### Формат выходных данных

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

### Примеры

стандартный ввод	стандартный вывод
6	0
0 1 1 2 3 3	1
5	1
4 1	0
1 4	0
3 6	
2 6	
6 5	
7	0
5 4 0 3 4 7 3	0
10	1
2 7	0
2 7	0
5 1	0
2 4	0
7 2	1
2 6	1
6 2	0
4 2	
5 1	
1 7	