

Задача А. Кратчайший путь коня

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

На шахматной доске размером 8×8 заданы две клетки. Соедините эти клетки кратчайшим путём коня.

Формат входных данных

Программа получает на вход координаты двух клеток, каждая в отдельной строке. Координаты клеток задаются в виде буквы (от «a» до «h») и цифры (от 1 до 8) без пробелов.

Формат выходных данных

Программа должна вывести путь коня, начинающийся и заканчивающийся в данных клетках и содержащий наименьшее число клеток.

Пример

стандартный ввод	стандартный вывод
a1 h8	a1 b3 c5 d7 e5 f7 h8

Задача В. Шайтан-машинка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	64 мегабайта

У Ибрагима есть магическая чёрная шайтан-машинка. На ней есть три кнопки и табло. Табло может показывать не более чем четырёхзначные числа. Каждая из кнопок меняет число некоторым образом: первая домножает его на 3, вторая прибавляет к нему сумму его цифр, а третья вычитает из него 2. В случае, если число становится отрицательным или превосходит 9 999, шайтан-машинка ломается. Ибрагим может нажимать кнопки в любом порядке. Он хочет узнать, как получить на табло число b после некоторой последовательности нажатий, если сейчас шайтан-машинка показывает a . Помогите ему найти минимальное необходимое число нажатий.

Формат входных данных

Единственная строка входного файла содержит два натуральных числа a и b , разделённых пробелом ($1 \leq a, b \leq 9\,999$).

Формат выходных данных

Выведите одно число — минимальное необходимое количество действий.

Примеры

стандартный ввод	стандартный вывод
14 45	3
18 12	3
14 29	2

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 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7

Задача D. Эвакуация

Имя входного файла: `stdin`
 Имя выходного файла: `stdout`
 Ограничение по времени: 3 секунды
 Ограничение по памяти: 64 мегабайта

Одна из Сверхсекретных организаций, чье название мы не имеем право разглашать, представляет собой сеть из N подземных бункеров, соединенных равными по длине туннелями, по которым из любого бункера можно добраться до любого другого (не обязательно напрямую). Связь с внешним миром осуществляется через специальные засекреченные выходы, которые расположены в некоторых из бункеров. Организации понадобилось составить план эвакуации персонала на случай экстренной ситуации. Для этого для каждого из бункеров необходимо узнать, сколько времени потребуется для того, чтобы добраться до ближайшего из выходов. Вам, как специалисту по таким задачам, поручено рассчитать необходимое время для каждого из бункеров по заданному описанию помещения Сверхсекретной организации. Для вашего же удобства бункеры занумерованы числами от 1 до N .

Формат входных данных

В первой строке записано число N , во второй — число K ($1 \leq N \leq 100\,000$, $1 \leq K \leq N$) — количество бункеров и количество выходов соответственно. Далее через пробел записаны K различных чисел от 1 до N , обозначающих номера бункеров, в которых расположены выходы. Потом идёт целое число M ($1 \leq M \leq 100\,000$) — количество туннелей. Далее вводятся M пар чисел — номера бункеров, соединенных туннелем. По каждому из туннелей можно двигаться в обе стороны. В организации не существует туннелей, ведущих из бункера в самого себя, зато может существовать более одного туннеля между парой бункеров.

Формат выходных данных

В первой строке выведите N чисел, разделённых пробелом — для каждого из бункеров минимальное время, необходимое чтобы добраться до выхода. Считайте, что время перемещения по одному туннелю равно 1. Во второй строке выведите N чисел — для каждого бункера номер ближайшего бункера с выходом, если таких несколько выведите бункер с наименьшим номером.

Примеры

stdin	stdout
3	1 0 1
1	2 2 2
2	
3	
1 2	
3 1	
2 3	

Задача Е. Дейкстра

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

Дан ориентированный взвешенный граф.

Найдите кратчайшее расстояние от одной заданной вершины до другой.

Формат входных данных

В первой строке входного файла 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 по N чисел — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое целое неотрицательное число, не превосходящее 10 000 — присутствие ребра данного веса. На главной диагонали матрицы всегда нули.

Формат выходных данных

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

Примеры

стандартный ввод	стандартный вывод
3 1 2 0 -1 2 3 0 -1 -1 4 0	6

Задача F. Кратчайший путь между вершинами

Имя входного файла: `dist.in`
Имя выходного файла: `dist.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Коль Дейкстру́ писать без кучи,
То тайм-лимит ты получишь...
А в совсем крутой задаче
Юзай кучу Фибоначчи!

Спектакль преподавателей ЛКШ.июль-2007

Дан неориентированный взвешенный граф. Требуется найти минимальный путь между двумя вершинами.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и рёбер графа соответственно ($1 \leq n \leq 100\,000$, $1 \leq m \leq 200\,000$). Вторая строка входного файла содержит натуральные числа s и t — номера вершин, длину пути между которыми требуется найти ($1 \leq s, t \leq n$, $s \neq t$).

Следующие m строк содержат описание рёбер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i , e_i и w_i — номерами концов ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 10\,000$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами s и t , или -1, если такого пути нет. Если путь есть, то вторая строка должна содержать одно целое неотрицательное число k — количество вершин в кратчайшем пути от s до t . В третьей строчке выведите k чисел — сам кратчайший путь. Если кратчайших путей несколько, выведите любой.

Примеры

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

Задача G. Многоэтажки и пожилая черепашка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Пожилая черепашка Полина очень любит прямоугольное клетчатое болото Осуждаево, в левом верхнем углу которого обитает уже много-много лет. Но вот незадача: многоэтажная застройка добралась и до её родного дома. Поэтому пожилая Полина решила подать жалобу в местные управляющие болотные органы. Но, к сожалению, здание правительства находится достаточно далеко от её привычного места обитания: в правом нижнем углу.

Долгие путешествия не пугают Полину, но буквально вчера она прочитала объявление, в котором подробно описывался план застройки Осуждаево: там были перечислены клетки болота в порядке их захвата многоэтажками; план составлен так, что каждая многоэтажка занимает ровно одну клетку. Полина — очень целеустремлённая пожилая черепашка, поэтому по пути к зданию правительства она умеет двигаться только либо вниз, либо вправо, совершая шаги, которые только приближают её к своей цели, но она не может находиться в клетке, которая захвачена многоэтажкой.

Помогите Полине узнать, какое максимальное количество многоэтажек может быть построено до того, как она потеряет возможность добраться до здания правительства и подать свою жалобу и осудить застройку Осуждаева.

Формат входных данных

В первой строке даны три целых числа n , m и k ($2 \leq n, m \leq 300$; $1 \leq k \leq n \cdot m - 1$) — размеры болота и количество клеток, которые будут застроены многоэтажками.

В следующих k строках даны по два целых числа x и y ($1 \leq x \leq n$; $1 \leq y \leq m$) — координаты клеток, которые будут последовательно застроены (в том порядке, в котором даны). Гарантируется, что все клетки различны. Также гарантируется, что стартовая клетка так и не будет застроена.

Формат выходных данных

В единственной строке выведите номер многоэтажки, после постройки которой пожилая Полина перестанет иметь возможность подать жалобу в правительство Осуждаево, или -1 , если этого не случится.

Примеры

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

Задача Н. Игрушечный лабиринт

Имя входного файла:	stdin
Имя выходного файла:	stdout
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Игрушечный лабиринт представляет собой прозрачную плоскую прямоугольную коробку, внутри которой есть препятствия и перемещается шарик. Коробку можно наклонять влево, вправо, к себе или от себя, после каждого наклона шарик перемещается в заданном направлении до ближайшего препятствия или до стенки лабиринта, после чего останавливается. Целью игры является загнать шарик в одно из специальных отверстий-выходов. Шарик проваливается в отверстие, если оно встречается на его пути.

Первоначально шарик находится в левом верхнем углу лабиринта. Гарантируется, что решение существует и левый верхний угол не занят препятствием или отверстием.

Формат входных данных

В первой строке входного файла записаны числа N и M — размеры лабиринта (целые положительные числа, не превышающие 100). Затем идёт N строк по M чисел в каждой — описание лабиринта. Число 0 в описании означает свободное место, число 1 — препятствие, число 2 — отверстие.

Например, лабиринту, изображённом на рисунке, будет соответствовать такое описание:

	4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0
--	---

Формат выходных данных

Выведите единственное число — минимальное количество наклонов, которые необходимо сделать, чтобы шарик покинул лабиринт через одно из отверстий.

Примеры

stdin	stdout
4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0	3

Задача I. Ожидая светофор

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

В городе имеется n перекрёстков и m направленных дорог.

На каждой из этих дорог стоит светофор, и для каждой дороги заданы два целых числа a и b :

- зеленый свет горит a секунд;
- красный свет горит b секунд;
- после этого снова горит зеленый a секунд, затем снова горит красный b секунд, и так далее.

Все светофоры начинают этот цикл одновременно в момент времени 0.

Автомобиль может въехать на дорогу только тогда, когда на светофоре горит зеленый свет, и обязан полностью покинуть её до того, как будет гореть красный.

На любом перекрёстке разрешается ожидать произвольное количество времени.

Необходимо определить минимальное время, за которое можно добраться из перекрёстка s в перекрёсток t .

Формат входных данных

Первая строка содержит четыре числа n, m, s, t ($1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5, 1 \leq s, t \leq n$). Далее следуют m строк. Каждая строка содержит пять целых чисел u, v, a, b, t , где

- u — начало дороги;
- v — конец дороги;
- a — время, в течение которого горит зеленый свет;
- b — время, в течение которого горит красный свет;
- t — время, необходимое для проезда по данной дороге.

Для всех дорог выполняются ограничения $1 \leq a, b, t \leq 10^9, 1 \leq u, v \leq n$. Гарантируется, что между одной и той же упорядоченной парой перекрёстков существует не более одной дороги. Между двумя перекрёстками могут существовать дороги в противоположных направлениях.

Формат выходных данных

Выведите минимальное время (в секундах), необходимое для того, чтобы добраться из перекрёстка s в перекрёсток t или -1 , если это невозможно.

Примеры

стандартный ввод	стандартный вывод
3 2 1 3 1 2 5 6 3 2 3 7 7 6	20
3 2 1 3 1 2 5 6 3 2 3 9 5 6	9