

## 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  $8 \times 8$  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<br>h8         | a1<br>b3<br>c5<br>d7<br>e5<br>f7<br>h8 |

## Задача В. 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<br>1 3<br>1 4<br>3 1<br>2 2<br>3 3 | 4                 |
| 3 4 5<br>1 3<br>1 4<br>3 1<br>2 4<br>3 2 | -1                |

## Задача С. 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 D. 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<br>0 -1 2<br>3 0 -1<br>-1 4 0 | 6                 |

## Задача Е. Лягушачьи тропы

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 2 секунды         |
| Ограничение по памяти:  | 512 мегабайт      |

На болоте живёт лягушка. Болото состоит из  $n$  кочек, между некоторыми из которых проложены тропы. По каждой тропе лягушка может прыгать в обе стороны, но на каждую тропу она тратит определённое количество сил. Лягушка хочет добраться с одной кочки на другую, потратив как можно меньше сил.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества кочек и троп на болоте соответственно ( $1 \leq n \leq 1000$ ,  $1 \leq m \leq \frac{n \cdot (n-1)}{2}$ ). Вторая строка входного файла содержит натуральные числа  $s$  и  $t$  — номера кочек, на которой лягушка сидит сейчас и на которую хочет попасть ( $1 \leq s, t \leq n$ ,  $s \neq t$ ).

Следующие  $m$  строк содержат описание троп по одной на строке. Тропа номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номерами кочек, которые она соединяет, и количеством сил, которое лягушка тратит на прыжок по этой тропе ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 10000$ ).

### Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — минимальное количество сил, которое лягушка потратит, чтобы добраться с кочки  $s$  на кочку  $t$ , или -1, если добраться невозможно. Если путь есть, то вторая строка должна содержать одно целое неотрицательное число  $k$  — количество кочек в кратчайшем маршруте лягушки от  $s$  до  $t$ . В третьей строчке выведите  $k$  чисел - сам маршрут (последовательность кочек). Если кратчайших маршрутов несколько, выведите любой.

### Пример

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

## Задача F. 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<br>0 0 0 0 1<br>0 1 1 0 2<br>0 2 1 0 0<br>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<br>0 0 0 0 1<br>0 1 1 0 2<br>0 2 1 0 0<br>0 0 1 0 0 | 3      |

## Задача G. Расстояние между вершинами

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Коль напишешь дейкстру ты - получишь реджект,  
беги, менты

Юрий 0 лет Зайцев

Дан неориентированный взвешенный граф. Требуется найти вес минимального пути 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 ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100$ ).

### 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.

### Примеры

| стандартный ввод                               | стандартный вывод |
|------------------------------------------------|-------------------|
| 4 4<br>1 3<br>1 2 1<br>2 3 2<br>3 4 5<br>4 1 4 | 3                 |

### Замечание

В эту задачу принимаются только решения с использованием алгоритма bfs.

## Задача Н. Кратчайший путь двух коней

Имя входного файла: knight2.in  
Имя выходного файла: knight2.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке.

### Формат входных данных

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

### Формат выходных данных

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений.

### Примеры

| knight2.in | knight2.out |
|------------|-------------|
| a1         | 1 b3        |
| c2         | 1 d4        |
| c2         | 2 a1        |
| a1         | 1 c2        |



## Задача I. Цивилизация

Имя входного файла: `civ.in`  
Имя выходного файла: `civ.out`  
Ограничение по времени: 0.5 секунда  
Ограничение по памяти: 256 мегабайт

Карта мира в компьютерной игре «Цивилизация» версии 1 представляет собой прямоугольник, разбитый на квадратики. Каждый квадратик может иметь один из нескольких возможных рельефов, для простоты ограничимся тремя видами рельефов — поле, лес и вода. Поселенец перемещается по карте, при этом на перемещение в клетку, занятую полем, необходима одна единица времени, на перемещение в лес — две единицы времени, а перемещаться в клетку с водой нельзя.

У вас есть один поселенец, вы определили место, где нужно построить город, чтобы как можно скорее завладеть всем миром. Найдите маршрут переселенца, по которому можно прийти в место строительства города за минимальное время. На каждом ходе переселенец может перемещаться в клетку, имеющую общую сторону с той клеткой, где он сейчас находится.

### Формат входных данных

Во входном файле записаны два натуральных числа  $N$  и  $M$ , не превосходящих 1000 — размеры карты мира ( $N$  — число строк в карте,  $M$  — число столбцов). Затем заданы координаты начального положения поселенца  $x$  и  $y$ , где  $x$  — номер строки,  $y$  — номер столбца на карте ( $1 \leq x \leq N$ ,  $1 \leq y \leq M$ ), строки нумеруются сверху вниз, столбцы — слева направо. Затем аналогично задаются координаты клетки, куда необходимо привести поселенца.

Далее идет описание карты мира в виде  $N$  строк, каждая из которых содержит  $M$  символов. Каждый символ может быть либо «.» (точка), обозначающим поле, либо «W», обозначающим лес, либо «#», обозначающим воду.

Гарантируется, что начальная и конечная клетки пути переселенца не являются водой.

### Формат выходных данных

В первой строке выходного файла выведите количество единиц времени, необходимое для перемещения поселенца (перемещение в клетку с полем занимает 1 единицу времени, перемещение в клетку с лесом — 2 единицы времени). Во второй строке выходного файла выведите последовательность символов, задающих маршрут переселенца. Каждый символ должен быть одним из четырех следующих: «N» (движение вверх), «E» (движение вправо), «S» (движение вниз), «W» (движение влево). Если таких маршрутов несколько — выведите любой из них.

Если дойти из начальной клетки в конечную невозможно, выведите число -1.

### Примеры

| civ.in                                                  | civ.out            |
|---------------------------------------------------------|--------------------|
| 4 8 1 1 4 8<br>...WWW<br>.#####.<br>.#..W...<br>...WWW. | 13<br>SSSEENEEEEES |
| 4 7 2 2 3 6<br>#####<br>#WW#..#<br>#WW#..#<br>#####     | -1                 |

## Задача К. Лабиринт

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

В одном из уровней компьютерной игры вы попали в лабиринт, состоящий из  $n$  строк, каждая из которых содержит  $m$  клеток. Каждая клетка либо свободна, либо занята препятствием. Стартовая клетка находится в строке  $r$  и столбце  $c$ . За один шаг вы можете переместиться на одну клетку вверх, влево, вниз или вправо, если она не занята препятствием. Вы не можете перемещаться за границы лабиринта.

К сожалению, ваша клавиатура крайне близка к поломке, поэтому вы можете переместиться влево не более  $x$  раз и вправо не более  $y$  раз. При этом ограничений на перемещения вверх и вниз нет, поскольку клавиши, используемые для движения вверх и вниз, всё ещё в идеальном состоянии.

Теперь вы для каждой клетки поля решили установить, можно ли выбрать такую последовательность нажатий, которая приведёт вас из стартовой в эту клетку. Посчитайте, сколько клеток поля обладают таким свойством.

### Формат входных данных

Первая строка содержит два целых числа  $n, m$  ( $1 \leq n, m \leq 2000$ ) — количество строк и столбцов в лабиринте, соответственно.

Вторая строка содержит два целых числа  $r, c$  ( $1 \leq r \leq n, 1 \leq c \leq m$ ) — номер строки и столбца, на пересечении которых расположена стартовая клетка.

Третья строка содержит два целых числа  $x, y$  ( $0 \leq x, y \leq 10^9$ ) — максимальное количество перемещений влево и вправо, соответственно.

Следующие  $n$  строк содержат описание лабиринта. Каждая из этих строк имеет длину  $m$  и состоит только из символов '.' и '\*'. В  $i$ -й строке  $j$ -й символ соответствует клетке лабиринта с номерами строки и столбца  $i$  и  $j$ , соответственно. Символ '.' соответствует свободной клетке лабиринта, а символ '\*' — клетке с препятствием.

Гарантируется, что стартовая клетка не занята препятствием.

### Формат выходных данных

Выведите одно число — количество клеток лабиринта, достижимых из стартовой, включая её саму.

### Примеры

| стандартный ввод                                               | стандартный вывод |
|----------------------------------------------------------------|-------------------|
| 4 5<br>3 2<br>1 2<br>.....<br>.***.<br>...**<br>*.....         | 10                |
| 5 5<br>5 4<br>3 1<br>**...<br>**.*.<br>...*.<br>.***.<br>..... | 16                |

## Задача L. Кратчайший путь между вершинами

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
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 ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 10000$ ).

### 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

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