

## Задача А. Кратчайший путь коня

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
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.

### Примеры

<code>stdin</code>	<code>stdout</code>
a1	a1
h8	b3
	c5
	d7
	e5
	f7
	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 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 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

<code>stdin</code>	<code>stdout</code>
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 0 -1 2 3 0 -1 -1 4 0	6

## Задача Е. Игрушечный лабиринт

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

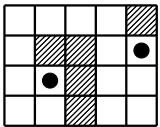
Игрушечный лабиринт представляет собой прозрачную плоскую прямоугольную коробку, внутри которой есть препятствия и перемещается шарик. Коробку можно наклонять влево, вправо, к себе или от себя, после каждого наклона шарик перемещается в заданном направлении до ближайшего препятствия или до стенки лабиринта, 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, изображённому на рисунке, будет соответствовать такое описание:

	<pre>4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0</pre>
--	--

### Формат выходных данных

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

### Примеры

<code>stdin</code>	<code>stdout</code>
<pre>4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0</pre>	<pre>3</pre>