## Задача A. Доставка кефирчика

| | |
|---|---|
| Имя входного файла: | `kefir.in` |
| Имя выходного файла: | `kefir.out` |
| Ограничение по времени: | 3 секунды |
| Ограничение по памяти: | 64 мегабайта |

Во время проведения очередной Межгалактической Летней Компьютерной Школы (МЛКШ) организаторы столкнулись с проблемой доставки кефирчика для вечерки. Дело в том, что кефирчик производят на планете под номером 1, а сами школьники живут на планете $n$, поэтому на доставку кефирчика тратится довольно большое время, а значит он успевает испортиться.

К счастью, галактическая транспортная система «Берендеев-Экспресс» постепенно внедряет новые кефиропроводы, способные передавать кефир со скоростью, в два раза превышающей скорость старых моделей. А именно, с любой планеты на любую по старым кефиропроводам кефир проходит за два года, а по новым — за один.

Разумеется, грешно было бы не воспользоваться инновационными технологиями, поэтому директор МЛКШ попросил вас написать программу, которая по данным о имеющихся кефиропроводах (как новых, так и старых) узнает кратчайший путь от планеты 1 до планеты $n$.

### Формат входных данных

В первой строке входного файла даны два целых числа $n$ и $m$ ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 100\,000$) — количество планет и количество кефиропроводов соответственно. В последующих $m$ строках даны тройки натуральных чисел $u_i$, $v_i$ и $c_i$. Числа $u_i$ и $v_i$ обозначают номера планет, соединенных $i$-м кефиропроводом, а $c_i$ ($c_i = 1$ или $c_i = 2$) — количество лет, которое потребуется, чтобы передать кефир с одной планеты на другую через $i$-й кефиропровод. Планеты во входном файле нумеруются с единицы. Кефир по трубопроводам можно передавать в обоих направлениях.

### Формат выходных данных

В выходной файл требуется вывести одно число — количество лет, которое требуется, чтобы доставить кефир с планеты 1 на планету $n$. Если доставка невозможна, то в выходной файл требуется вывести «`-1`».

### Примеры

| kefir.in | kefir.out |
|---|---|
| 3 2<br>1 2 2<br>2 3 1 | 3 |
| 3 1<br>2 3 1 | -1 |
| 2 5<br>1 2 1<br>1 2 2<br>1 2 1<br>1 1 2<br>2 2 1 | 1 |

## Задача B. Эвакуация

| | |
|---|---|
| Имя входного файла: | `evacuation.in` |
| Имя выходного файла: | `evacuation.out` |
| 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 тунелями, 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 $K$ ($1 \leqslant N \leqslant 100\,000$, $1 \leqslant K \leqslant N$) — 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 $M$ ($1 \leqslant M \leqslant 100\,000$) — количество туннелей. Далее вводятся $M$ пар чисел — номера бункеров, соединенных туннелем. По каждому из тунелей 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

| evacuation.in | evacuation.out |
|---|---|
| 3<br>1<br>2<br>3<br>1 2<br>3 1<br>2 3 | 1 0 1<br>2 2 2 |

## Задача C. Числа

| | |
|---|---|
| Имя входного файла: | numbers.in |
| Имя выходного файла: | numbers.out |
| Ограничение по времени: | 1 секунда |
| Ограничение по памяти: | 64 мегабайта |

Витя хочет придумать новую игру с числами. В этой игре от игроков требуется преобразовывать четырехзначные числа не содержащие нулей при помощи следующего разрешенного набора действий:

1. Можно увеличить первую цифру числа на 1, если она не равна 9.
2. Можно уменьшить последнюю цифру на 1, если она не равна 1.
3. Можно циклически сдвинуть все цифры на одну вправо.
4. Можно циклически сдвинуть все цифры на одну влево.

Например, применяя эти правила к числу 1234, можно получить числа 2234, 1233, 4123 и 2341 соответственно.

Точные правила игры Витя пока не придумал, но пока его интересует вопрос, как получить из одного числа другое за минимальное количество операций.

### Формат входных данных

Во входном файле содержится два различных четырехзначных числа, каждое из которых не содержит нулей.

### Формат выходных данных

Программа должна вывести последовательность четырехзначных чисел, не содержащих нулей. Последовательность должна начинаться первым из данных чисел и заканчиваться вторым из данных чисел, каждое последующее число в последовательности должно быть получено из предыдущего числа применением одного из правил. Количество чисел в последовательности должно быть минимально возможным.

### Примеры

| numbers.in | numbers.out |
|---|---|
| 9876<br>8876 | 9876<br>8769<br>8768<br>8876 |

## Задача D. Кратчайшее расстояние

| | |
|---|---|
| Имя входного файла: | mindist.in |
| Имя выходного файла: | mindist.out |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

Дан ориентированный граф. Найдите расстояния от вершины $x$ до всех остальных вершин графа.

### Формат входных данных

В первой строке входного файла содержатся два натуральных числа $N$ и $x$ ($1 \leqslant N \leqslant 1\,000$, $1 \leqslant x \leqslant N$) — количество вершин в графе и стартовая вершина соответственно. Далее в $N$ строках по $N$ чисел — матрица смежности графа: в $i$-й строке на $j$-м месте стоит «1», если вершины $i$ и $j$ соединены ребром, и «0», если ребра между ними нет. На главной диагонали матрицы стоят нули.

### Формат выходных данных

Выведите через пробел числа $d_1, d_2, \ldots d_n$, где $d_i$ — это $-1$, если путей между $x$ и $i$ нет, и минимальное расстояние между $x$ и $i$ в противном случае.

### Примеры

| mindist.in | mindist.out |
|---|---|
| 6 5<br>0 1 1 0 0 0<br>1 0 0 0 0 0<br>1 1 0 0 0 0<br>0 0 0 0 1 0<br>0 0 1 1 0 0<br>0 1 0 0 0 0 | 2 2 1 1 0 -1 |

## Задача E. Приключения шахматного коня

| | |
|---|---|
| Имя входного файла: | knight.in |
| Имя выходного файла: | knight.out |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 64 мегабайта |

На шахматной доске $N \times N$ в клетке $(x1, y1)$ стоит голодный шахматный конь. Он хочет попасть в клетку $(x2, y2)$, где растет вкусная шахматная трава. Какое наименьшее количество ходов он должен для этого сделать?

### Формат входных данных

На вход программы поступает пять чисел: $N$, $x1$, $y1$, $x2$, $y2$ ($5 \leqslant N \leqslant 20$, $1 \leqslant x1, y1, x2, y2 \leqslant N$). Левая верхняя клетка доски имеет координаты $(1, 1)$, правая нижняя - $(N, N)$.

### Формат выходных данных

В первой строке выведите единственное число $K$ - наименьшее необходимое число ходов коня. В каждой из следующих $K + 1$ строк должно быть записано 2 числа - координаты очередной клетки в пути коня.

### Примеры

| knight.in | knight.out |
|---|---|
| 5<br>1 1<br>3 2 | 2<br>1 1<br>3 2 |

## Задача F. Обход в глубину

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

Дан неориентированный невзвешенный граф, в котором выделена вершина. Вам необходимо найти количество вершин, лежащих с ней в одной компоненте связности (включая саму выделенную вершину).

### Формат входных данных

В первой строке входного файла содержатся два целых числа $N$ и $S$ ($1 \leqslant S \leqslant N \leqslant 100$), где $N$ — количество вершин графа, а $S$ — выделенная вершина. В следующих $N$ строках записано по $N$ чисел — матрица смежности графа, в которой цифра «`0`» означает отсутствие ребра между вершинами, а цифра «`1`» — его наличие. Гарантируется, что на главной диагонали матрицы всегда стоят нули.

### Формат выходных данных

Выведите одно целое число — искомое количество вершин.

### Примеры

| dfs.in | dfs.out |
|---|---|
| 5 1<br>0 1 1 0 0<br>1 0 1 0 0<br>1 1 0 0 0<br>0 0 0 0 1<br>0 0 0 1 0 | 3 |