

## Задача А. 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 задается 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 В. Рейсы во времени

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

Между  $N$  населёнными пунктами совершаются пассажирские рейсы на машинах времени.

В момент времени 0 вы находитесь в пункте  $A$ . Вам дано расписание рейсов. Требуется оказаться в пункте  $B$  как можно раньше (то есть в наименьший возможный момент времени).

При этом разрешается делать пересадки с одного рейса на другой. Если вы прибываете в некоторый пункт в момент времени  $T$ , то вы можете уехать из него любым рейсом, который отправляется из этого пункта в момент времени  $T$  или позднее (но не раньше).

### Формат входных данных

Первая строка входного файла содержит число  $N$  — количество населённых пунктов ( $1 \leq N \leq 1000$ ). Вторая строка содержит два числа  $A$  и  $B$  — номера начального и конечного пунктов. Третья строка содержит число  $K$  — количество рейсов ( $0 \leq K \leq 1000$ ). Следующие  $K$  строк содержат описания рейсов, по одному на строке. Каждое описание представляет собой четвёрку целых чисел. Первое число каждой четвёрки задаёт номер пункта отправления, второе — время отправления, третье — пункт назначения, четвёртое — время прибытия. Номера пунктов — натуральные числа из диапазона от 1 до  $N$ . Пункт назначения и пункт отправления могут совпадать. Время измеряется в некоторых абсолютных единицах и задаётся целым числом, по модулю не превышающим  $10^9$ . Поскольку рейсы совершаются на машинах времени, то время прибытия может быть как больше времени отправления, так и меньше, или равным ему.

Гарантируется, что входные данные таковы, что добраться из пункта  $A$  в пункт  $B$  всегда можно.

### Формат выходных данных

Выведите в выходной файл минимальное время, когда вы сможете оказаться в пункте  $B$ .

### Примеры

<code>stdin</code>	<code>stdout</code>
2 1 1 2 1 1 2 10 1 10 1 9	0
1 1 1 3 1 3 1 -5 1 -5 1 -3 1 -4 1 -10	-10

## Задача С. 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