

## Задача А. 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 В. 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 задаётся 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 С. После финала

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

Финал Чемпионата Бйгтландии по программированию впервые проводился в городе  $\mathcal{I}$ . Дорожная сеть города  $\mathcal{I}$  представляет собой  $N$  перекрёстков, соединённых  $M$  дорогами с двусторонним движением. Программа финала была столь насыщенной, что участники не успели осмотреть город. Более того, церемония закрытия затянулась, так что при отъезде команды-победителя времени оставалось только на то, чтобы добраться до аэропорта кратчайшим по суммарной длине путём. При этом, чтобы хотя бы немного посмотреть город, из всех таких маршрутов выбрали тот, который включает в себя целиком наибольшее количество дорог.

По заданной карте города вычислите длину маршрута и количество дорог, которое удалось посмотреть команде.

### Формат входных данных

Первая строка содержит два целых числа  $N$  и  $M$  — количество перекрёстков в городе  $\mathcal{I}$  и количество дорог соответственно ( $2 \leq N \leq 1000$ ,  $1 \leq M \leq 2 \cdot 10^5$ ).

Каждая из следующих  $M$  строк содержит три целых числа  $a_i$ ,  $b_i$  и  $c_i$  ( $1 \leq a_i, b_i \leq N$ ,  $a_i \neq b_i$ ,  $1 \leq c_i \leq 1000$ ) — номера перекрёстков, соединяемых  $i$ -й дорогой, и длину этой дороги. Два различных перекрёстка могут быть соединены более чем одной дорогой. Место проведения финала находится на перекрёстке с номером 1, аэропорт — на перекрёстке с номером  $N$ .

Гарантируется, что существует хотя бы один путь от места проведения финала в аэропорт.

### Формат выходных данных

Выведите два целых числа  $P$  и  $Q$  — длину кратчайшего пути от места проведения финала в аэропорт и максимальное число дорог, которые может включать в себя кратчайший путь.

### Примеры

<code>excursion.in</code>	<code>excursion.out</code>
3 3 1 2 1 1 3 2 2 3 1	2 2

## Задача D. Авиаперелёты-2

Имя входного файла: avia2.in  
Имя выходного файла: avia2.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Профессору Форду необходимо попасть на международную конференцию. Он хочет потратить на дорогу наименьшее количество денег, поэтому решил, что будет путешествовать исключительно ночными авиарейсами (чтобы не тратиться на ночевку в отелях), а днем будет осматривать достопримечательности тех городов, через которые он будет проезжать транзитом. Он внимательно изучил расписание авиаперелетов и выбрал подходящие ему авиарейсы, выяснив, что перелеты на выбранных направлениях совершаются каждую ночь и за одну ночь он не сможет совершить два перелета.

Теперь профессор хочет найти путь наименьшей стоимости, учитывая что до конференции осталось  $K$  ночей (то есть профессор может совершить не более  $K$  перелетов).

### Формат входных данных

В первой строке находятся числа  $N$  (количество городов),  $M$  (количество авиарейсов),  $K$  (количество оставшихся ночей),  $S$  (номер города, в котором живет профессор),  $F$  (номер города, в котором проводится конференция). Ограничения:  $2 \leq N \leq 100$ ,  $1 \leq M \leq 10^5$ ,  $1 \leq K \leq 100$ ,  $1 \leq S \leq N$ ,  $1 \leq F \leq N$ .

Далее идет  $M$  строк, задающих расписание авиарейсов.  $i$ -я строка содержит три натуральных числа:  $S_i$ ,  $F_i$  и  $P_i$ , где  $S_i$  — номер города, из которого вылетает  $i$ -й рейс,  $F_i$  — номер города, в который прилетает  $i$ -й рейс,  $P_i$  — стоимость перелета  $i$ -м рейсом.  $1 \leq S_i \leq N$ ,  $1 \leq F_i \leq N$ ,  $1 \leq P_i \leq 10^6$ .

### Формат выходных данных

Выведите одно число — минимальную стоимость пути, подходящего для профессора. Если профессор не сможет за  $K$  ночей добраться до конференции, выведите число -1.

### Примеры

avia2.in	avia2.out
4 5 2 1 4 1 2 1 2 3 1 3 4 1 1 3 3 1 4 5	4

## Задача Е. Заправки

Имя входного файла: `gasstation.in`  
Имя выходного файла: `gasstation.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В стране  $N$  городов, некоторые из которых соединены между собой дорогами. Для того, чтобы проехать по одной дороге, требуется один бак бензина. В каждом городе бак бензина имеет разную стоимость. Вам требуется добраться из первого города в  $N$ -й, потратив как можно меньшее количество денег.

Дополнительно имеется канистра для бензина, куда входит столько же бензина, сколько входит в бак. В каждом городе можно заправить бак, залить бензин в канистру, залить и туда и туда, или же перелить бензин из канистры в бак.

### Формат входных данных

В первой строке дано число  $N$  ( $1 \leq N \leq 100$ ). Во второй строке входного файла записано  $N$  чисел,  $i$ -е из которых задает стоимость бензина в  $i$ -м городе (все числа целые в диапазоне от 0 до 100).

Во следующих строках описаны все дороги (по одной в строке). Каждая дорога задается двумя числами — номерами городов, которые она соединяет. Все дороги двухсторонние, между двумя городами существует не более одной дороги, не существует дорог, ведущих из города в себя.

### Формат выходных данных

В выходной файл выведите одно число — суммарную стоимость маршрута или  $-1$ , если добраться до нужного города невозможно.

### Примеры

<code>gasstation.in</code>	<code>gasstation.out</code>
4 1 10 2 15 1 2 1 3 4 2 4 3	2

## Задача F. 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