

## Задача А. Путь в графе

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

В неориентированном графе требуется найти длину минимального пути между двумя вершинами.

### Формат входных данных

Первым на вход поступает число  $N$  — количество вершин в графе ( $1 \leq N \leq 100$ ). Затем записана матрица смежности (0 обозначает отсутствие ребра, 1 — наличие ребра). Далее задаются номера двух вершин — начальной и конечной.

### Формат выходных данных

Необходимо вывести длину пути в ребрах. Если пути нет, нужно вывести  $-1$ .

### Пример

стандартный ввод	стандартный вывод
5 0 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 3 5	3

## Задача В. Кратчайший и быстрый путь между вершинами

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

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.

### Пример

<code>dist.in</code>	<code>dist.out</code>
4 4	3
1 3	3
1 2 1	1 2 3
2 3 2	
3 4 5	
4 1 4	

## Задача С. Пифагоров экспресс (для C++)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2.5 секунд  
Ограничение по памяти: 256 мегабайт

В стране Флатландии есть  $n$  городов, расположенных в целочисленных точках плоскости. Транспортная система Флатландии настолько развита, что между любыми двумя городами ходит экспресс имени Пифагора. С помощью него можно добраться от города с координатами  $x_1, y_1$  до города с координатами  $x_2, y_2$  за время  $(x_1 - x_2)^2 + (y_1 - y_2)^2$ . По каждой линии ходит достаточно поездов, и временем на пересадки можно пренебречь.

Сообщение с внешним миром во Флатландии продумано несколько хуже, и единственный аэропорт международного сообщения находится в городе с номером  $s$ . Вам же хочется попасть в город с номером  $t$ . Определите, за какое наименьшее время это можно сделать.

### Формат входных данных

В первой строке задано число городов  $n$  ( $1 \leq n \leq 2 \cdot 10^4$ ). В следующих  $n$  строках заданы координаты городов  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^4$ ). В последней строке даны  $s$  и  $t$  — номера начального и конечного города в пути ( $1 \leq s, t \leq n$ ).

### Формат выходных данных

Выведите одно число — минимальное время, за которое можно добраться из  $s$  в  $t$ .

### Пример

стандартный ввод	стандартный вывод
3 0 1 0 0 1 3 2 3	6

## Задача D. Флойд

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

Дан ориентированный граф, в котором могут быть кратные ребра и петли. Каждое ребро имеет вес, выражающийся целым числом (возможно, отрицательным). Гарантируется, что циклы отрицательного веса отсутствуют.

Требуется посчитать матрицу длин кратчайших путей от каждой вершины до всех остальных вершин.

### Формат входных данных

Программа получает сначала число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа и число  $M$  ( $0 \leq M \leq 10^4$ ) — количество ребер.

В следующих строках идет  $M$  троек чисел, описывающих ребра: начало ребра, конец ребра и вес (вес — целое число от  $-100$  до  $100$ ).

### Формат выходных данных

Требуется вывести матрицу  $N$  на  $N$ . В  $j$ -м столбце  $i$ -й строки следует вывести длину кратчайшего пути от вершины  $i$  до вершины  $j$ , либо число 30 000, если не существует пути между данными вершинами.

### Пример

стандартный ввод	стандартный вывод
6 4	0 10 20 30000 30000 30000
1 2 10	30000 0 10 30000 30000 30000
2 3 10	30000 30000 0 30000 30000 30000
1 3 100	30000 30000 30000 0 -10 30000
4 5 -10	30000 30000 30000 30000 0 30000
	30000 30000 30000 30000 30000 0

## Задача Е. Один голодный конь

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

На шахматной доске  $N \times N$  в клетке  $(x_1, y_1)$  стоит голодный шахматный конь. Он хочет попасть в клетку  $(x_2, y_2)$ , где растет вкусная шахматная трава. Какое наименьшее количество ходов он должен для этого сделать?

### Формат входных данных

На вход программы поступает пять чисел:  $N, x_1, y_1, x_2, y_2$  ( $5 \leq N \leq 20, 1 \leq x_1, y_1, x_2, y_2 \leq N$ ).  
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.

### Пример

стандартный ввод	стандартный вывод
5	1
1 1	1 1
3 2	3 2

## Задача F. Числа

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

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:

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

### Пример

стандартный ввод	стандартный вывод
1234	1234
4321	2234
	3234
	4323
	4322
	4321

## Задача G. Игрушечный лабиринт

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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. Лабиринт 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 (шарик не обязан останавливаться в отверстии).

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 идет  $N$  строк по  $M$  чисел в каждой — описание лабиринта. Число 0 в описании означает свободное место, число 1 — препятствие, число 2 — отверстие.

### Формат выходных данных

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

### Пример

стандартный ввод	стандартный вывод
4 5 0 0 0 0 1 0 1 1 0 2 0 2 1 0 0 0 0 1 0 0	3

## Задача Н. Полнейший Флойд

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

Вам дан **ориентированный** граф. Если в нём есть отрицательные циклы, то выведите хотя бы один из них. В противном случае выведите матрицу попарных кратчайших расстояний между вершинами.

### Формат входных данных

На первой строке даны  $1 \leq V \leq 500$  — число вершин в графе и  $0 \leq E \leq \min(2.5 \times 10^4, |V| \times |V|)$  — число рёбер в графе. Далее записаны рёбра графа, по одному в строке. Каждое ребро записано в виде трёх целых чисел:  $1 \leq from, to \leq |V|, |w| \leq 10^5$ .

### Формат выходных данных

На первой строк выведите LOOP или NO LOOP в зависимости от наличия отрицательного цикла.

Далее, если в графе есть отрицательный цикл, выведите на второй строке его длину, а на третьей строке выведите его вершины в любом порядке, соблюдая направление рёбер графа. В противном случае выведите  $V$  строк по  $V$  чисел, где  $j$ -й элемент на  $i$ -й строке означает кратчайшее расстояние между вершинами  $i$  и  $j$  и  $INF$  означает, что пути между вершинами нет.

### Примеры

стандартный ввод	стандартный вывод
3 3 1 2 1 2 3 2 1 3 2	NO LOOP 0 1 2 INF 0 2 INF INF 0
3 3 1 2 -1 2 3 -1 3 1 -1	LOOP 3 3 1 2
7 7 2 1 -4 2 7 -17 3 1 -14 4 5 16 4 5 -19 4 5 14 7 6 -15	NO LOOP 0 INF INF INF INF INF INF -4 0 INF INF INF -32 -17 -14 INF 0 INF INF INF INF INF INF INF 0 -19 INF INF INF INF INF INF 0 INF INF INF INF INF INF INF 0 INF INF INF INF INF INF -15 0



## Задача I. Пятница, тринадцатое

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Девочка Маша очень суеверная, поэтому, когда наступает пятница, тринадцатое, она начинает вести себя очень неадекватно. К сожалению, ей нужно ходить в школу, и ей приходится ездить на автобусах. В городе есть  $N$  автобусных остановок и  $M$  автобусных маршрутов. Если в пятницу, тринадцатое она проедет от какой-то остановки до какой-то другой (возможно, используя несколько маршрутов) за время  $T$ , причём  $T$  делится на 13, то она начинает истошно орать и бегать по автобусу. Помогите ей добраться до школы и остаться в здравом уме.

### Формат входных данных

В первой строке входного файла задано одно число  $T$  ( $1 \leq T \leq 10$ ) — число тестов. В первой строке теста заданы два числа  $N$  и  $M$  ( $1 \leq N \leq 50$ ,  $1 \leq M \leq 2500$ ) — число остановок и маршрутов соответственно. Следующие  $M$  строк описывают маршруты в формате From To Time ( $1 \leq \text{From}, \text{To} \leq N$ ,  $1 \leq \text{Time} \leq 100$ ) — откуда и куда едет автобус и время в пути. На последней строчке теста будет указано, является ли сегодняшний день пятницей, тринадцатым (True) или нет (False).

### Формат выходных данных

Для каждого теста выведите на отдельной строчке минимальное время, за которое Маша сможет доехать от дома (остановка 1) до школы (остановка  $N$ ), или  $-1$ , если она этого сделать не сможет.

### Пример

стандартный ввод	стандартный вывод
3	16
5 5	-1
1 2 1	42
1 3 2	
2 4 1	
3 4 3	
4 5 11	
True	
2 1	
1 2 26	
True	
3 3	
1 1 7	
1 2 26	
2 3 16	
False	

## Задача J. Найти короткий путь

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

Однажды Юрий решил снарядить очередной поход в горы и собрался взять с собой несколько друзей. Горы можно представить как  $n$  полян, между которыми есть тропы. Юрий знает тропы двух типов — короткие и длинные. По каждой тропе можно ходить только в одном направлении. Правда, длинные тропы не всегда длиннее коротких.

Для того, чтобы друзья не считали своего проводника скучным человеком, Юрий всегда чередует тропы. Строго говоря, Юрий никогда не ходит по двум тропам одного типа подряд. Юрий рассматривает несколько маршрутов через горы, каждый из которых заканчивается на поляне с номером  $n$ , от которой можно добраться до города. Не смотря на длину троп, каждую из них Юрий проходит за одну минуту. Для каждого варианта начальной поляны посчитайте минимальное время прогулки от текущей поляны до поляны с номером  $n$ , если Юрий верен своим принципам и никогда не ходит по двум тропам одного типа подряд.

### Формат входных данных

В первой строке через пробел заданы два числа  $n$  и  $m$  — количество полян и троп соответственно ( $2 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$ ).

В следующих  $m$  строках записаны три числа:  $a_i, b_i$  и  $type_i$ , означающие, что с поляны  $a_i$  на поляну  $b_i$  ведет тропа типа  $type_i$ , где  $type_i \in \{1, 2\}$  ( $type_i$  означает тип тропы).

Обратите внимание, что тропа может соединять поляну саму с собой, а также между двумя полянами может быть несколько троп.

### Формат выходных данных

Выведите через пробел  $n - 1$  число — минимальное время прогулки от вершины с номером  $i$  до вершины с номером  $n$ . Если от какой-то вершины нельзя добраться до вершины с номером  $n$ , выполняя принципы Юрия, выведите вместо минимального времени число  $-1$ .

### Пример

стандартный ввод	стандартный вывод
6 6 1 2 1 2 3 2 3 1 1 1 6 2 4 3 1 5 1 2	1 3 2 -1 5

## Задача К. Олимпиада по алхимии

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

В государстве алхимиков есть  $N$  населённых пунктов, пронумерованных числами от 1 до  $N$ , и  $M$  дорог. Населённые пункты бывают двух типов: деревни и города. Кроме того, в государстве есть одна столица (она может располагаться как в городе, так и в деревне). Каждая дорога соединяет два населённых пункта, и для проезда по ней требуется  $T_i$  минут. В столице было решено провести 1-ю государственную командную олимпиаду по алхимии. Для этого во все города из столицы были отправлены гонцы (по одному гонцу на город) с информацией про олимпиаду.

Напишите программу, которая посчитает, в каком порядке и через какое время каждый из гонцов доберётся до своего города. Считается, что гонец во время пути не спит и нигде не задерживается.

### Формат входных данных

Во входном файле сначала записаны 3 числа  $N$ ,  $M$ ,  $K$  — количество населённых пунктов, количество дорог и количество городов ( $2 \leq N \leq 1000$ ,  $1 \leq M \leq 10000$ ,  $1 \leq K \leq N$ ). Далее записан номер столицы  $C$  ( $1 \leq C \leq N$ ).

Следующие  $K$  чисел задают номера городов.

Далее следуют  $M$  троек чисел  $S_i$ ,  $E_i$ ,  $T_i$ , описывающих дороги:  $S_i$  и  $E_i$  — номера населённых пунктов, которые соединяет данная дорога, а  $T_i$  — время для проезда по ней ( $1 \leq T_i \leq 100$ ).

Гарантируется, что до каждого города из столицы можно добраться по дорогам (возможно, через другие населённые пункты).

### Формат выходных данных

Выведите в выходной файл  $K$  пар чисел: для каждого города должен быть выведен его номер и минимальное время, когда гонец может в нем оказаться (время измеряется в минутах с того момента, как гонцы выехали из столицы). Пары в выходном файле должны быть упорядочены по времени прибытия гонца.

### Примеры

стандартный ввод	стандартный вывод
5 4 5 1 1 2 3 4 5 1 2 1 2 3 10 3 4 100 4 5 100	1 0 2 1 3 11 4 111 5 211
5 5 3 1 2 4 5 2 1 1 2 3 10 3 4 100 4 5 100 1 5 1	2 1 5 1 4 101

## Задача L. Заправки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
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  $N$ -ый, потратив как можно меньше денег. Покупать бензин впрок нельзя.

### Формат входных данных

В первой строке вводится число  $N$  ( $1 \leq N \leq 100$ ), в следующей строке идет  $N$  чисел,  $i$ -ое из которых задает стоимость бензина в  $i$ -ом городе (всё это целые числа из диапазона от 0 до 100).

Затем идет число  $M$  — количество дорог в стране, далее идет описание самих дорог. Каждая дорога задается двумя числами — номерами городов, которые она соединяет. Все дороги двухсторонние (то есть по ним можно ездить как в одну, так и в другую сторону), между двумя городами всегда существует не более одной дороги, не существует дорог, ведущих из города в себя.

### Формат выходных данных

Требуется вывести одно число — суммарную стоимость маршрута или  $-1$ , если добраться невозможно.

### Примеры

стандартный ввод	стандартный вывод
5 3 6 1 7 6 8 1 2 5 4 5 1 3 4 5 2 2 4 2 3 3 1	3
5 3 7 2 9 4 4 1 2 1 3 2 3 4 5	-1