

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 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 В. Лабиринт знаний (25 баллов)

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

В ЛКШ построили аттракцион «Лабиринт знаний». Лабиринт представляет собой N комнат, занумерованных от 1 до N , между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определённую величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход — в комнате N . Каждый ЛКШонок проходит лабиринт ровно один раз и попадает в группу в зависимости от набранных знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача показать наилучший результат.

Формат входных данных

Первая строка входного файла содержит целые числа N ($1 \leq N \leq 2000$) — количество комнат и M ($0 \leq M \leq 10000$) — количество дверей. В каждой из следующих M строк содержится описание двери — номера комнат, из которой она ведёт и в которую она ведёт, а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10000). Двери могут вести из комнаты в неё саму, между двумя комнатами может быть более одной двери.

Формат выходных данных

В выходной файл выведите «:)» — если можно получить неограниченно большой запас знаний, «:(» — если лабиринт пройти нельзя, и максимальное количество набранных знаний в противном случае.

Примеры

<code>maze.in</code>	<code>maze.out</code>
2 2 1 2 3 1 2 7	7

Задача С. Цикл отрицательного веса

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

Дан ориентированный граф. Определите, есть ли в нем цикл отрицательного веса, и если да, то выведите его.

Формат входных данных

Во входном файле в первой строке число N ($1 \leq N \leq 100$) — количество вершин графа. В следующих N строках находится по N чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

Формат выходных данных

В первой строке выходного файла выведите «YES», если цикл существует или «NO» в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле и в третьей строке — вершины входящие в этот цикл в порядке обхода.

Примеры

<code>negcycle.in</code>	<code>negcycle.out</code>
2	YES
0 -1	2
-1 0	2 1

Задача D. Кратчайшие пути

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

Вам дан взвешенный ориентированный граф и вершина s в нём. Для каждой вершины графа u выведите длину кратчайшего пути от вершины s до вершины u .

Формат входных данных

Первая строка входного файла содержит три целых числа n , m , s — количество вершин и рёбер в графе и номер начальной вершины соответственно ($2 \leq n \leq 2\,000$, $1 \leq m \leq 5\,000$).

Следующие m строчек описывают рёбра графа. Каждое ребро задаётся тремя числами — начальной вершиной, конечной вершиной и весом ребра соответственно. Вес ребра — целое число, не превосходящее 10^{15} по абсолютной величине. В графе могут быть кратные рёбра и петли.

Формат выходных данных

Выведите n строчек — для каждой вершины u выведите длину кратчайшего пути из s в u . Если не существует пути между s и u , выведите «*». Если не существует кратчайшего пути между s и u , выведите «-».

Примеры

path.in	path.out
6 7 1	0
1 2 10	10
2 3 5	-
1 3 100	-
3 5 7	-
5 4 10	*
4 3 -18	
6 1 -1	

Задача Е. Диаметр графа

Имя входного файла: `diameter.in`
Имя выходного файла: `diameter.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 16 мегабайта

Дан связный взвешенный неориентированный граф.

Рассмотрим пару вершин, расстояние между которыми максимально среди всех пар вершин.

- Расстояние (длина кратчайшего пути) между ними называется *диаметром графа*.
- *Эксцентриситетом вершины v* называется максимальное из расстояний от вершины v до других вершин графа.
- *Радиусом графа* называется наименьший из эксцентриситетов вершин.

Найдите диаметр и радиус графа.

Формат входных данных

В первой строке входных данных содержится единственное число: n ($1 \leq n \leq 100$) — количество вершин графа. В следующих n строках по n чисел — матрица длин ребер графа, где -1 означает отсутствие ребра между вершинами, а любое целое неотрицательное число, не превосходящее $1/000$ — присутствие ребра данного веса. На главной диагонали матрицы всегда нули.

Гарантируется, что матрица симметрична, а граф — связный.

Формат выходных данных

Выведите два числа — диаметр и радиус графа.

Примеры

<code>diameter.in</code>	<code>diameter.out</code>
4	8
0 -1 1 2	5
-1 0 -1 5	
1 -1 0 4	
2 5 4 0	

Задача F. Заправки

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

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 входного файла записано N чисел ($1 \leq N \leq 25$), 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>
1 10 2 15	2
1 2	
1 3	
4 2	
4 3	

Задача G. 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 Н. Егор и граф

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

У Егора есть взвешенный ориентированный граф, состоящий из n вершин. В этом графе между любой парой различных вершин есть ребро в обоих направлениях. Егор любит играть с графом, и сейчас он придумал новую игру:

- Игра состоит из n шагов.
- На i -м шаге Егор удаляет из графа вершину номер x_i . Удаляя вершину, Егор удаляет все ребра, которые входили в данную вершину и которые выходили из нее.
- Перед выполнением каждого шага, Егор хочет знать сумму длин кратчайших путей между всеми парами оставшихся вершин. Кратчайший путь может проходить через любую оставшуюся вершину. Другими словами, если обозначить как $d(i, v, u)$ кратчайший путь между вершинами v и u в графе, который получился до удаления вершины x_i , то Егор хочет знать значение следующей суммы:
$$\sum_{v, u, v \neq u} d(i, v, u).$$

Помогите Егору, выведите значение искомой суммы перед каждым шагом.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 500$) — количество вершин в графе.

В следующих n строках содержится по n целых чисел — матрица смежности графа: j -е число в i -й строке a_{ij} ($1 \leq a_{ij} \leq 10^5$, $a_{ii} = 0$) обозначает вес ребра, ведущего из вершины i в вершину j .

В следующей строке содержится n различных целых чисел: x_1, x_2, \dots, x_n ($1 \leq x_i \leq n$) — вершины, которые удаляет Егор.

Формат выходных данных

Выведите n целых чисел — i -е число равно искомой сумме перед i -м шагом.

Примеры

stdin	stdout
2	9 0
0 5	
4 0	
1 2	

Задача I. 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 J. 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 К. Pink Floyd

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

Группа Pink Floyd собирается отправиться в новый концертный тур по всему миру. По предыдущему опыту группа знает, что солист Роджер Уотерс постоянно нервничает при перелетах. На некоторых маршрутах он теряет вес от волнения, а на других — много ест и набирает вес.

Известно, что чем больше весит Роджер, тем лучше выступает группа, поэтому требуется спланировать перелеты так, чтобы вес Роджера на каждом концерте был максимально возможным.

Группа должна посещать города в том же порядке, в котором она дает концерты. При этом между концертами группа может посещать промежуточные города.

Формат входных данных

Первая строка входного файла содержит три натуральных числа n , m и k — количество городов в мире, количество рейсов и количество концертов, которые должна дать группа соответственно ($n \leq 100$, $m \leq 10\,000$, $2 \leq k \leq 10\,000$). Города пронумерованы числами от 1 до n .

Следующие m строк содержат описание рейсов, по одному на строке. Рейс номер i описывается тремя числами b_i , e_i и w_i — номер начального и конечного города рейса и предполагаемое изменение веса Роджера в миллиграммах ($1 \leq b_i, e_i \leq n$, $-100\,000 \leq w_i \leq 100\,000$).

Последняя строка содержит числа a_1, a_2, \dots, a_k — номера городов, в которых проводятся концерты ($a_i \neq a_{i+1}$). В начале концертного тура группа находится в городе a_1 .

Гарантируется, что группа может дать все концерты.

Формат выходных данных

Первая строка выходного файла должна содержать число l — количество рейсов, которые должна сделать группа. Вторая строка должна содержать l чисел — номера используемых рейсов.

Если существует такая последовательность маршрутов между концертами, что Роджер будет набирать вес неограниченно, то первая строка выходного файла должна содержать строку “infinitely kind”.

Примеры

floyd.in	floyd.out
4 8 5	6
1 2 -2	5 6 5 7 2 3
2 3 3	
3 4 -5	
4 1 3	
1 3 2	
3 1 -2	
3 2 -3	
2 4 -10	
1 3 1 2 4	

Задача L. 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.

Примеры

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 М. В поисках ученика

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

Один непослушный ученик ЛКШ решил сбежать с пар! Разумеется, он был пойман, однако преподавателям стало интересно, какой путь преодолел ученик.

В ЛКШ есть n домиков, пронумерованных от 1 до n . Некоторые пары домиков соединены тропинками. По каждой тропинке можно перемещаться в две стороны. У каждой тропинки есть длина, выражающаяся целым числом метров. Всего есть m тропинок, соединяющих домики. Система тропинок очень сложная и запутанная, поэтому может быть так, что пара домиков соединена более, чем одной тропинкой. Сбежавший ученик начал свой путь в домике с номером s , пробежал вдоль нескольких тропинок и закончил путь в домике с номером t . Также ученик рассказал, что до некоторого момента он бегал только вдоль тропинок длиной не более A метров, чтобы поскорее прибегать к своим друзьям в новом домике, а после этого он бегал только по тропинкам длиной не менее B метров, чтобы преподаватели не могли его догнать.

Преподаватели поняли, что обладая этой информацией, возможно, не удастся восстановить путь ученика однозначно, однако им стало интересно, какой кратчайший путь он мог преодолеть. Найдите путь из домика s в домик t минимальной суммарной длины, который сначала идет только по тропинкам длины не более A , а потом только по тропинками длины хотя бы B .

Формат входных данных

Первая строка входных данных содержит два целых числа n и m — количество домиков и тропинок в ЛКШ ($1 \leq n, m \leq 10^5$). Вторая строка содержит два целых числа s и t — номера начального и конечного домика в пути ($1 \leq s, t \leq n$). Каждая из следующих m строк содержит по три целых числа a , b и c , описывающих тропинку, соединяющую домики с номерами a и b , имеющую длину c ($1 \leq a, b \leq n$, $1 \leq c \leq 10^9$). Последняя строка содержит два целых числа A и B ($1 \leq A, B \leq 10^9$).

Формат выходных данных

Если ни одного возможного пути ученика не существует, выведите единственное число -1 . В противном случае, выведите единственное число — минимальную возможную длину пути ученика.

Примеры

path.in	path.out
4 6 1 4 1 2 4 2 4 4 4 1 6 1 3 7 4 3 10 3 2 2 5 6	6
6 9 2 4 2 6 4 6 4 5 4 5 1 5 3 10 3 2 9 2 5 5 1 5 2 1 6 2 1 2 1 5 3	4