

Задача А. Автобусы

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

Между некоторыми деревнями края Васюки ходят автобусы. Поскольку пассажиропотоки здесь не очень большие, то автобусы ходят всего несколько раз в день.

Марии Ивановне требуется добраться из деревни d в деревню v как можно быстрее (считается, что в момент времени 0 она находится в деревне d).

Формат входных данных

Сначала вводится число N — общее число деревень ($1 \leq N \leq 100$), затем номера деревень d и v , за ними следует количество автобусных рейсов R ($0 \leq R \leq 10000$). Далее идут описания автобусных рейсов. Каждый рейс задается номером деревни отправления, временем отправления, деревней назначения и временем прибытия (все времена — целые от 0 до 10000). Если в момент t пассажир приезжает в какую-то деревню, то уехать из нее он может в любой момент времени, начиная с t .

Формат выходных данных

Выведите минимальное время, когда Мария Ивановна может оказаться в деревне v . Если она не сможет с помощью указанных автобусных рейсов добраться из d в v , выведите -1 .

Пример

стандартный ввод	стандартный вывод
3 1 3 4 1 0 2 5 1 1 2 3 2 3 3 5 1 1 3 10	5

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.

Обратите внимание на файловый ввод/вывод.

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

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

Задача С. Заправки-2

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. Помимо этого у вас есть канистра для бензина, куда входит столько же топлива, сколько входит в бензобак.

В каждом городе бак бензина имеет разную стоимость. Вам требуется добраться из первого города в n -й, потратив как можно меньше денег.

В каждом городе можно заправить бак, заправить бак и канистру или же перелить бензин из канистры в бак. Это позволяет экономить деньги, покупая бензин в тех городах, где он стоит дешевле, но канистры хватает только на одну заправку бака!

Формат входных данных

В первой строке вводится число n ($1 \leq n \leq 100$), в следующей строке идет n чисел, i -е из которых задает стоимость бензина в i -м 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.

Пример

стандартный ввод	стандартный вывод
4 1 10 2 15 4 1 2 1 3 4 2 4 3	2

Задача D. Мистер Флойд

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

Дан ориентированный взвешенный граф. Вам необходимо найти пару вершин, кратчайшее расстояние от одной из которых до другой максимально среди всех пар вершин.

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, где -1 означает отсутствие ребра между вершинами, а любое неотрицательное число – присутствие ребра данного веса. На главной диагонали матрицы – всегда нули.

Формат выходных данных

Выведите искомое максимальное кратчайшее расстояние.

Пример

стандартный ввод	стандартный вывод
6 0 6 8 -1 -1 -1 5 0 5 -1 -1 -1 1 7 0 -1 -1 -1 -1 -1 -1 0 6 -1 -1 -1 -1 -1 0 3 -1 -1 -1 2 -1 0	9

Задача E. Флойд - существование

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

Дан ориентированный взвешенный граф. По его матрице смежности нужно для каждой пары вершин определить, существует ли кратчайший путь между ними или нет.

Формат входных данных

В первой строке входного файла записано единственное число n ($1 \leq n \leq 100$) — количество вершин графа. В следующих n строках по n чисел — матрица смежности графа (j -е число в i -й строке соответствует весу ребра из вершины i в вершину j): число 0 обозначает отсутствие ребра, а любое другое число — наличие ребра соответствующего веса. Все числа по модулю не превышают 100.

Формат выходных данных

Выведите n строк по n чисел. j -е число в i -й строке должно соответствовать кратчайшему пути из вершины i в вершину j . Число должно быть равно 0, если пути не существует, 1, если существует кратчайший путь, и 2, если пути существуют, но бывают пути сколь угодно маленького веса.

Пример

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

Замечание

Кратчайший путь может не существовать по двум причинам:

- нет ни одного пути
- есть пути сколь угодно маленького веса

Задача F. Транспортировка

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

К очередной Летней компьютерной школе было решено подготовить кружки как для школьников, так и для всех преподавателей.

Имея привычку делать важные дела в самый последний момент, дизайнер закончил работу над макетом за два дня до начала школы. Ещё день уйдёт у завода-изготовителя на то, чтобы изготовить кружки и нанести на них изображение. На то, чтобы довезти кружки от завода-изготовителя до ЛКШ, остаётся всего 24 часа.

Заказ на 10000000 экземпляров кружек (а именно столько заказали организаторы), конечно же, за один рейс не увезти. Однако, за первый рейс хочется привезти максимальное количество кружек. Для перевозки был заказан один большегрузный автомобиль. Но есть один нюанс: на некоторых дорогах установлено ограничение на вес автомобиля. Поэтому если автомобиль нагрузить кружками под завязку, то, возможно, не удастся воспользоваться самым коротким маршрутом, а придётся ехать в объезд. Может случиться даже так, что из-за этого грузовик не успеет доехать до лагеря вовремя, а этого допустить никак нельзя. Итак, сколько же кружек можно погрузить в автомобиль, чтобы успеть привезти этот ценный груз вовремя, и не нарушая правил дорожного движения?

Формат входных данных

В первой строке находятся числа n ($1 \leq n \leq 500$) и m - количество узловых пунктов дорожной схемы и количество дорог, соответственно. В следующих m строках находится информация о дорогах. Каждая дорога описывается в отдельной строке следующим образом. Сначала указаны номера узловых пунктов, которые соединяются данной дорогой, потом время, которое тратится на проезд по этой дороге, и, наконец, максимальный вес автомобиля, которому разрешено ехать по этой дороге. Известно, что все дороги соединяют различные пункты, причем для каждой пары пунктов есть не более одной дороги, непосредственно их соединяющей. Все числа разделены одним или несколькими пробелами.

Узловые пункты нумеруются числами от 1 до n . При этом завод по производству кружек имеет номер 1, а ЛКШ - номер n . Время проезда по дороге задано в минутах и не превосходит 1440 (24 часа). Ограничение на массу задано в граммах и не превосходит одного миллиарда. Кроме того, известно, что одна кружка весит 100 грамм, а пустой грузовик - 3 тонны.

Формат выходных данных

Выведите одно число - максимальное количество кружек, которое можно привезти за первый рейс, потратив не более 24 часов.

Пример

стандартный ввод	стандартный вывод
3 3 1 2 10 3000220 2 3 20 3000201 1 3 1 3000099	2

Задача G. Защищенное соединение

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

В свете недавних новостей о прослушке каналов связи, два непримиримых интернет-гиганта Урагании «Laim.UR» и «Xenda» решили подписать соглашение об установлении защищенного канала связи между дата-центрами друг друга. В Урагании n городов, но, к сожалению, ни в одном городе нет дата-центров обоих гигантов. Поэтому для формирования защищенного канала придется прокладывать междугородные линии связи.

Специалисты компаний определили m пар городов, которые можно соединить, проложив сегмент канала связи, и оценили стоимость создания такого сегмента для каждой из этих пар.

Результирующий канал может состоять из нескольких сегментов. Он должен начинаться в одном из городов, где находится дата-центр первой компании, может проходить через промежуточные города и должен заканчиваться в городе, где находится дата-центр второй компании.

Теперь необходимо определить минимальную стоимость защищенного канала, соединяющего два дата-центра компаний.

Формат входных данных

В первой строке находятся целые числа n и m ($2 \leq n \leq 5000$, $1 \leq m \leq 10^5$) — количество городов и количество пар городов, которые можно соединить сегментом канала связи.

Во второй строке находятся n целых чисел a_i ($0 \leq a_i \leq 2$). Если $a_i = 0$, то в i -м городе нет дата-центра ни одного из гигантов. Если $a_i = 1$, то в i -м городе есть дата-центр «Laim.UR», а если $a_i = 2$, то в i -м городе находится дата-центр «Xenda». Гарантируется, что среди этих чисел есть как минимум одна единица и одна двойка.

В каждой из следующих m строк находится по три целых числа — s_i , t_i и c_i , которые означают, что города s_i и t_i ($1 \leq s_i, t_i \leq n$, $s_i \neq t_i$) можно соединить сегментом канала связи стоимостью c_i ($1 \leq c_i \leq 10^5$). Каждую пару городов можно соединить не более чем одним сегментом канала.

Формат выходных данных

Если соединить защищенным каналом связи два дата-центра разных интернет-гигантов возможно, то выведите в выходной файл три числа: x , y и d , означающие, что между городами x и y возможно провести канал связи суммарной стоимостью d . В городе x должен находиться дата-центр «Laim.UR», в городе y — дата-центр «Xenda». Если существует несколько оптимальных ответов, выведите любой. Если провести искомый канал невозможно, выведите -1.

Примеры

стандартный ввод	стандартный вывод
6 7 1 0 1 2 2 0 1 3 3 1 2 4 2 3 3 2 4 2 1 6 5 3 5 6 5 6 1	3 4 5
4 2 1 0 0 2 1 3 3 2 4 2	-1