

Задача А. Авиаперелёты

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

Главного конструктора Петю попросили разработать новую модель самолёта для компании «Air Бубундия». Оказалось, что самая сложная часть заключается в подборе оптимального размера топливного бака.

Главный картограф «Air Бубундия» Вася составил подробную карту Бубундии. На этой карте он отметил расход топлива для перелёта между каждой парой городов.

Петя хочет сделать размер бака минимально возможным, для которого самолёт сможет долететь от любого города в любой другой (возможно, с дозаправками в пути).

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 1000$) — число городов в Бубундии. Далее идут n строк по n чисел каждая. j -е число в i -й строке равно расходу топлива при перелёте из i -го города в j -й. Все числа не меньше нуля и меньше 10^9 . Гарантируется, что для любого i в i -й строчке i -е число равно нулю.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — оптимальный размер бака.

Примеры

avia.in	avia.out
4 0 10 12 16 11 0 8 9 10 13 0 22 13 10 17 0	10

Задача В. Опасный маршрут

Имя входного файла: `danger.in`
Имя выходного файла: `danger.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Профессор Дейкстра живёт в очень опасном районе города. Ежедневно бандиты грабят на улицах прохожих. Читая криминальную хронику, профессор вычислил вероятность быть ограбленным при проходе по каждой улице города.

Теперь он хочет найти наиболее безопасный путь от дома до университета, в котором он преподаёт. Иными словами, он хочет найти путь от дома до университета, для которого вероятность быть ограбленным минимальна.

Формат входных данных

В первой строке входного файла записаны два числа N и M — количество зданий и количество улиц, соединяющих здания ($1 \leq N \leq 100$, $1 \leq M \leq N(N-1)/2$). В следующей строке находятся числа S и F — номер дома, в котором живёт профессор и номер дома, в котором находится университет соответственно. Далее в M строках расположены описания дорог: 3 целых числа S_i , F_i и P_i — номера зданий, в которых начинается и заканчивается дорога, и вероятность в процентах быть ограбленным, пройдя по дороге соответственно ($1 \leq S_i \leq N$, $1 \leq F_i \leq N$, $0 \leq P_i \leq 100$, дороги двунаправленные). Гарантируется, что существует хотя бы один путь от дома профессора до университета.

Формат выходных данных

Необходимо вывести одно число — минимальную возможную вероятность быть ограбленным с точностью не менее 6 знаков после запятой.

Примеры

<code>danger.in</code>	<code>danger.out</code>
3 3 1 3 1 2 20 1 3 50 2 3 20	0.359999999999999996

Задача С. Транспортировка

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

Компания “Яндекс” решила подарить всем школьникам и преподавателям ЛКШ оригинальные кружки. К сожалению, количество необходимых кружек оказалось столь велико, что изготовитель доставил кружки в офис Яндекса в самый последний момент. До открытия смены в "Берендеевых полянах" осталось всего 24 часа.

О плачевном состоянии дорог по пути на базу ходят легенды — в частности, на многих разбитых дорогах действует ограничение на вес автомобиля. Соответственно, от нагруженности машины зависит возможность воспользоваться тем или иным маршрутом, тяжёлой машине может потребоваться ехать в обход.

Уже совершенно очевидно, что все кружки не поспеют к открытию. Чтобы спасти ситуацию, отвезите первым рейсом максимально возможное количество кружек успев до начала открытия смены.

Формат входных данных

В первой строке находятся целые числа n ($2 \leq n \leq 500$) и m — количество городов и количество двусторонних дорог, соответственно.

В следующих m строках описываются дороги.

В каждой строке находятся целые числа a_i, b_i, t_i, w_i — соответственно два города, ею соединяемые, время на проезд по ней в минутах и ограничение на вес автомобиля в граммах ($t_i \leq 1440$, $w_i \leq 10^9$, $1 \leq a_i, b_i \leq n$)

Между каждой парой городов есть не более одной дороги.

Кроме того, известно, что офис Яндекса имеет номер 1, а “Берендеевы поляны” — номер n , одна кружка весит 100 грамм, а пустой грузовик — 3 тонны.

Формат выходных данных

Выведите одно число — максимальное количество кружек, которое можно привезти, потратив не более 24 часов.

Примеры

<code>cups.in</code>	<code>cups.out</code>
3 3 1 2 10 3000220 2 3 20 3000201 1 3 1 3000099	2

Задача D. Транспортировка (версия для python)

Имя входного файла: `cups.in`
Имя выходного файла: `cups.out`
Ограничение по времени: 6 секунды
Ограничение по памяти: 256 мегабайт

Компания “Яндекс” решила подарить всем школьникам и преподавателям ЛКШ оригинальные кружки. К сожалению, количество необходимых кружек оказалось столь велико, что изготовитель доставил кружки в офис Яндекса в самый последний момент. До открытия смены в “Берендеевых полянах” осталось всего 24 часа.

О плачевном состоянии дорог по пути на базу ходят легенды — в частности, на многих разбитых дорогах действует ограничение на вес автомобиля. Соответственно, от нагруженности машины зависит возможность воспользоваться тем или иным маршрутом, тяжёлой машине может потребоваться ехать в обход.

Уже совершенно очевидно, что все кружки не поспеют к открытию. Чтобы спасти ситуацию, отвезите первым рейсом максимально возможное количество кружек успев до начала открытия смены.

Формат входных данных

В первой строке находятся целые числа n ($2 \leq n \leq 100$) и m — количество городов и количество двусторонних дорог, соответственно.

В следующих m строках описываются дороги.

В каждой строке находятся целые числа a_i, b_i, t_i, w_i — соответственно два города, ею соединяемые, время на проезд по ней в минутах и ограничение на вес автомобиля в граммах ($t_i \leq 1440$, $w_i \leq 10^9$, $1 \leq a_i, b_i \leq n$)

Между каждой парой городов есть не более одной дороги.

Кроме того, известно, что офис Яндекса имеет номер 1, а “Берендеевы поляны” — номер n , одна кружка весит 100 грамм, а пустой грузовик — 3 тонны.

Формат выходных данных

Выведите одно число — максимальное количество кружек, которое можно привезти, потратив не более 24 часов.

Примеры

<code>cups.in</code>	<code>cups.out</code>
3 3 1 2 10 3000220 2 3 20 3000201 1 3 1 3000099	2

Задача Е. Наименьшее кратное

Имя входного файла: `multiple.in`
Имя выходного файла: `multiple.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дано число X и множество цифр D . Требуется дописать к X минимальное количество цифр из D , чтобы получившееся число делилось на k . При этом получившееся число должно быть минимально возможным.

Формат входных данных

Первая строка входного файла содержит два натуральных числа X и k ($1 \leq X \leq 10^{1000}, 2 \leq k \leq 100\,000$). Во второй строке записано количество цифр во множестве D . В третьей строке через пробел записаны эти цифры.

Формат выходных данных

Единственная строка должна содержать минимальное число, полученное из X дописыванием цифр из D и кратное k . Если такого числа не существует, выведите -1.

Примеры

<code>multiple.in</code>	<code>multiple.out</code>
102 101 3 1 0 3	10201
202 101 3 1 0 3	202

Задача F. Pink Floyd

Имя входного файла: floyd.in
Имя выходного файла: floyd.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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 G. Болото

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Недавно в непосредственной близости от ЛКШ, как раз на месте предполагаемого посвящения, в лесу было обнаружено болото. Для того, чтобы никто из ЛКШат не пострадал, на исследование области был отправлен робот. Ваша задача — написать программу, управляющую роботом, для нахождения его площади.

Поле покрыто сеткой с квадратными ячейками одинакового размера. Болото представляет собой четырехсвязную область, то есть из каждой его клетки можно попасть в другую, переходя лишь через стороны квадратов. Внутри болота нет незаболоченных клеток. Размер болота по каждой координате ограничен 30-ю клетками. Каждая из клеток может быть свободной или заболоченной. Робот начинает работу в свободной клетке, соседней с болотом. Он может перемещаться на юг, запад, север или восток в свободную клетку. Робот не имеет оптических сенсоров, так что только при попытке перемещения в заболоченную клетку сработает механизм и робот останется в той же клетке. При этом вы получаете сообщение о том, что клетка относится к болоту.

Протокол интерактивного взаимодействия

Программа должна выводить на стандартный вывод одну строку с действием робота и ждать строки в стандартном вводе с ответом, затем выводить очередную строку с действием и считать ответ и так далее. Проходимые клетки могут быть посещены несколько раз. Для завершения процесса требуется вывести одно целое число — площадь болота.

Формат выходных данных

Каждая строка выходных данных должна представлять собой команду для робота или вывод результата. Всего существует четыре команды управления роботом: **SOUTH**, **WEST**, **NORTH** или **EAST**. После вывода площади программа должна завершать свою работу. Вывод каждой команды обязательно должен завершаться выводом конца строки.

Формат входных данных

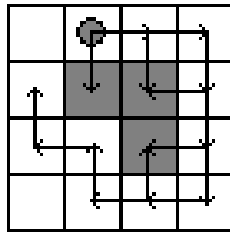
Каждая строка входных данных представляет собой ответ на действие робота. Это может быть строка **EMPTY**, если робот успешно переместился в заданном направлении, или строка **SWAMP**, если робот не смог переместиться из-за того, что клетка, в которую он хотел попасть, заболочена.

Пример

стандартный ввод	стандартный вывод
SWAMP	SOUTH
EMPTY	EAST
SWAMP	SOUTH
EMPTY	EAST
EMPTY	SOUTH
SWAMP	WEST
EMPTY	SOUTH
SWAMP	WEST
EMPTY	SOUTH
EMPTY	WEST
SWAMP	NORTH
EMPTY	WEST
EMPTY	NORTH
EMPTY	WEST
EMPTY	NORTH
	3

Замечание

Иллюстрация к примеру:



В точности соблюдайте формат выходных данных. После вывода каждой строки сбрасывайте буфер вывода — для этого используйте команды `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

Задача Н. Авиаперелёты-2

Имя входного файла: avia2.in
 Имя выходного файла: avia2.out
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 64 мегабайта

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