

Задача А. Опасный маршрут

Имя входного файла: `danger.in`
Имя выходного файла: `danger.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

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 ($1 \leq N \leq 500$, $1 \leq M \leq N(N-1)/2$). 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	0.36000000000000004219

Задача В. 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

cups.in	cups.out
3 3	2
1 2 10 3000220	
2 3 20 3000201	
1 3 1 3000099	

Задача С. 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 D. Сеть дорог

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

Лондон — очень старинный город. По ходу истории город разрастался, и в нем появлялись новые дороги. Сейчас дороги в Лондоне бывают двух типов: кольцевые и радиальные. Количество кольцевых дорог равно k . Пронумеруем их от 1 до k , тогда i -я кольцевая дорога представляет собой квадрат со сторонами параллельными осям координат, и с противоположными углами в точках $(-i, -i)$ и (i, i) . Любая радиальная дорога является отрезком и соединяет две кольцевые дороги с номерами, отличающимися ровно на 1, или кольцевую дорогу с номером 1 и точку $(0, 0)$. При этом, она не имеет других общих точек с кольцевыми дорогами, кроме своих концов. Никакие две радиальные дороги не пересекаются, но могут иметь общие концы.

Паддингтон хочет добраться из точки a с координатами (x_a, y_a) в точку b с координатами (x_b, y_b) , перемещаясь только по дорогам. Найдите минимальное расстояние, которое ему придется преодолеть.

Смотрите пояснение к тестам из примера для лучшего понимания условия.

Формат входных данных

В первой строке дано два целых числа n и k — количество радиальных дорог и количество кольцевых дорог ($0 \leq n \leq 10^5$, $1 \leq k \leq 10^9$). В следующих n строках дано по 4 целых числа x_{i1} , y_{i1} , x_{i2} и y_{i2} — координаты начала и конца i -й радиальной дороги ($-k \leq x_{i1}, y_{i1}, x_{i2}, y_{i2} \leq k$). В последней строке дано четыре целых числа x_a , y_a , x_b , y_b — координаты точек a и b , соответственно ($-10^9 \leq x_a, y_a, x_b, y_b \leq 10^9$).

Гарантируется, что дороги удовлетворяют ограничениям, описанным в условии.

Формат выходных данных

Выведите одно вещественное число с абсолютной или относительной погрешностью не более 10^{-6} — кратчайшее расстояние из точки a до точки b . Или -1 , если пути не существует.

Система оценки

Эта задача состоит из пяти подзадач. Для подзадач выполняются дополнительные ограничения, указанные в таблице ниже. Для получения баллов за подзадачу необходимо пройти все тесты данной подзадачи, а также все тесты всех необходимых подзадач. Необходимые подзадачи также указаны в таблице.

Обратите внимание, что **тест из условия** не подходит под ограничения некоторых подзадач, однако он обязательно **должен быть пройден** для того, чтобы решение было принято на проверку.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	12	$n = 0, k \leq 1000$	
2	22	$k \leq 30, n \leq 500$. Каждая радиальная дорога соединяет либо углы квадратов, либо центры сторон квадратов. Посмотрите пояснение к третьему тесту из примера для лучшего понимания.	
3	22	$k \leq 200, n \leq 500$	2
4	22	$k \leq 50000$	2, 3
5	22		2, 3, 4

Примеры

стандартный ввод	стандартный вывод
1 1 0 0 1 1 -1 -1 0 0	5.414213562373095
5 3 1 0 2 2 2 1 3 2 -1 1 0 2 -1 -2 0 -1 -3 -1 -2 -1 -3 -1 3 2	10.414213562373096
5 3 0 0 1 1 3 -3 2 -2 2 0 1 0 0 -1 0 0 -2 0 -3 0 1 3 0 0	14.828427124746192

Задача Е. 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 F. Горилла и лабиринт

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

Это интерактивная задача.

Ваша цель – написать программу, управляющую гориллом, идущим вслепую по лабиринту.

Лабиринт состоит из n на m ($1 \leq n, m \leq 30$) клеток. Каждая из клеток может быть свободной или заблокированной (непроходимой). Все клетки на границе лабиринта непроходимые.

Горилл начинает в свободной клетке, он может переместиться на **юг**, **запад**, **север** или **восток** в свободную клетку.

У горилла завязаны глаза, но он вытянул вперед руки, так что при попытке перемещения в заблокированную клетку сработает инстинкт самосохранения и горилл останется в той же клетке.

Горилл должен побывать во всех проходимых клетках лабиринта. Из начальной клетки гарантированно можно попасть во все достижимые клетки лабиринта.

Формат входных данных

Каждая строка выходных данных должна представлять собой команду для горилла.

Это должна быть одна из пяти возможных строк: SOUTH, WEST, NORTH, EAST, или DONE.

Строка DONE должна быть напечатана после посещения роботом всех проходимых клеток. После вывода DONE программа должна немедленно завершать свою работу.

Необходимо очищать выходной буфер после вывода каждой команды (flush) или использовать endl.

Протокол взаимодействия

Программа должна выводить на стандартный вывод одну строку с действием горилла и ждать строки в стандартном вводе с ответом, затем выводить очередную строку с действием и считывать ответ и так далее до тех пор, пока все проходимые клетки лабиринта не будут посещены.

Программа должна завершать работу только когда все проходимые клетки будут посещены. Проходимые клетки могут быть посещены несколько раз. Допустимо передвигаться даже если все проходимые клетки уже посещены.

Пример

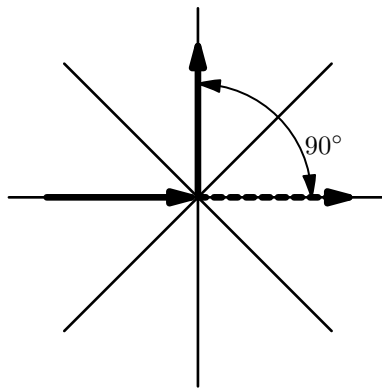
стандартный ввод	стандартный вывод
#####	EAST
#..o#	NORTH
##..#	WEST
#####	NORTH
	WEST
	NORTH
	WEST
	SOUTH
	EAST
	SOUTH
	EAST
	EAST
	SOUTH
	WEST
	WEST
	SOUTH
	NORTH
	EAST
	DONE

Задача G. Безопасный путь

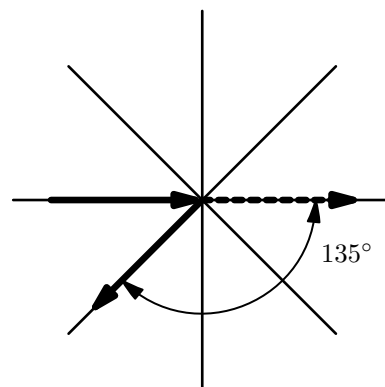
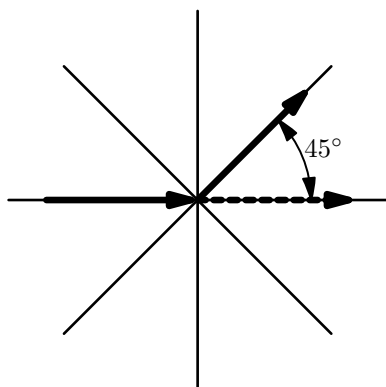
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя и Вася — хорошие друзья. Поэтому они часто ездят друг к другу в гости. Недавно Петя получил водительские права и собирается навестить своего друга. Для простоты будем считать, что все дороги в городе, в котором они живут, являются бесконечными прямыми. В месте пересечения двух или более дорог находятся перекрестки. Дома Пети и Васи расположены возле некоторых дорог города, но не на перекрестках.

Петя начинает путь на дороге возле своего дома. При этом он может выбрать любое из двух направлений. Когда Петя подъезжает к перекрестку, он может повернуть на любую другую проходящую через него дорогу или продолжить ехать по текущей. Поскольку Петя не очень опытный водитель, каждый поворот, который он совершает, заставляет его волноваться. Причем волнение Пети равно величине угла, на который он поворачивает, в градусах. Например, при повороте на прямой угол волнение Пети равно 90.



При менее крутом повороте Петя волнуется меньше, а при более крутом — сильнее.



Будем считать, что волнение Пети в течении всего маршрута равно сумме величин в градусах углов, на которые ему придется повернуть в течении движения. Конечно, Петя хочет воспользоваться маршрутом, который заставит его волноваться как можно меньше.

Помогите Пете выяснить, чему равно минимальное суммарное волнение, которое он испытает, добравшись до дома Васи.

Формат входных данных

В первой строке входного файла находится целое число n ($1 \leq n \leq 50$) — количество дорог в городе. В следующих n строках находится описание дорог.

Каждая дорога описывается четверкой целых чисел x_1, y_1, x_2, y_2 , которые задают координатами двух различных точек (x_1, y_1) и (x_2, y_2) , через которые проходит дорога.

Гарантируется, что никакие две дороги не совпадают. В следующих двух строках заданы координаты домов Пети и Васи. Гарантируется, что каждый дом находится ровно на одной дороге, а также, что Петя и Вася живут в разных местах.

Координаты всех точек во входном файле являются целыми числами и не превосходят 100 по абсолютному значению.

Формат выходных данных

В выходной файл выведите единственное число — суммарный угол в градусах, на который придется повернуть Пете при оптимальном выборе маршрута. Ответ считается правильным, если его относительная или абсолютная погрешность не превосходит 10^{-9} .

Если Петя никак не сможет добраться до дома Васи, выведите число -1.

Примеры

стандартный ввод	стандартный вывод
3 0 0 2 0 1 1 0 2 1 2 3 2 -3 0 3 2	270.000000000000000000
1 0 0 2 0 0 0 2 0	0.000000000000000000

Замечание

Следующий рисунок соответствует первому примеру. Петя совершает два поворота на 135 градусов, его суммарное волнение равно 270.

