,

Задача A. Лабиринт

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

В одном из уровней компьютерной игры вы попали в лабиринт, состоящий из n строк, каждая из которых содержит m клеток. Каждая клетка либо свободна, либо занята препятствием. Стартовая клетка находится в строке r и столбце c. За один шаг вы можете переместиться на одну клетку вверх, влево, вниз или вправо, если она не занята препятствием. Вы не можете перемещаться за границы лабиринта.

К сожалению, ваша клавиатура крайне близка к поломке, поэтому вы можете переместиться влево не более x раз и вправо не более y раз. При этом ограничений на перемещения вверх и вниз нет, поскольку клавиши, используемые для движения вверх и вниз, всё ещё в идеальном состоянии.

Теперь вы для каждой клетки поля решили установить, можно ли выбрать такую последовательность нажатий, которая приведёт вас из стартовой в эту клетку. Посчитайте, сколько клеток поля обладают таким свойством.

Формат входных данных

Первая строка содержит два целых числа n, m ($1 \leqslant n, m \leqslant 2000$) — количество строк и столбцов в лабиринте, соответственно.

Вторая строка содержит два целых числа r, c ($1 \leqslant r \leqslant n, 1 \leqslant c \leqslant m$) — номер строки и столбца, на пересечении которых расположена стартовая клетка.

Третья строка содержит два целых числа x, y ($0 \leqslant x, y \leqslant 10^9$) — максимальное количество перемещений влево и вправо, соответственно.

Следующие n строк содержат описание лабиринта. Каждая из этих строк имеет длину m и состоит только из символов '.' и '*'. В i-й строке j-й символ соответствует клетке лабиринта с номерами строки и столбца i и j, соответственно. Символ '.' соответствует свободной клетке лабиринта, а символ '*' — клетке с препятствием.

Гарантируется, что стартовая клетка не занята препятствием.

Формат выходных данных

Выведите одно число — количество клеток лабиринта, достижимых из стартовой, включая её саму.

Пример

стандартный ввод	стандартный вывод
4 5 3 2 1 2 ..... .***. ...** *....	10

Задача B. Робинзон и крокодилы

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

Робинзон живет на острове, который представляет собой прямоугольник размером $n \times m$ клеток.

На остров Робинзона выползли погреться на солнышке и задремали несколько крокодилов. Робинзон хочет прогнать неприятных соседей, не поднимая шума. Для этого он кидает в дремлющих крокодилов орехи.

В каждой клетке острова находится не более одного крокодила. Напуганный орехом крокодил быстро бежит строго по прямой, пока не окажется в воде. Для каждого крокодила известно направление, в котором он побежит, если его напугать. Направления, в которых будут убегать крокодилы, параллельны сторонам острова.

Если на пути напуганного крокодила окажется другой крокодил, то, столкнувшись, они разозлятся, и нападут на Робинзона. Поэтому надо тщательно выбирать очередного крокодила, чтобы на его пути были только пустые клетки.

Робинзон не кидает очередной орех, пока предыдущий крокодил не окажется в воде.

Требуется написать программу, определяющую максимальное количество крокодилов, которых можно прогнать, не разозлив их.

Формат входных данных

В первой строке входного файла записаны числа n и m — размеры острова с севера на юг и с запада на восток. Последующие n строк по m символов в каждой описывают текущее расположение крокодилов на острове. Если клетка свободна, то она обозначается точкой «`.`», а если там находится крокодил, то в ней указано направление, в котором побежит этот крокодил. Направления обозначаются буквами: «`N`» — север, «`S`» — юг, «`E`» — восток, «`W`» — запад.

Формат выходных данных

Выходной файл должен содержать одно число — максимальное количество крокодилов, которых можно прогнать, не разозлив.

Примеры

стандартный ввод	стандартный вывод
1 1 .	0
1 1 W	1
5 7 ....S... ..WE... ...N...	2
2 2 ES NW	0

Задача C. Недалекий Маршалл [B', B]

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

В Нью-Йорке есть n перекрестков и m улиц между ними. Перекрестки, которые соединены улицей, называются соседними.

Лили и Маршалл очень любят друг друга и поэтому хотят все время быть на соседних перекрестках. При этом, чтобы не отвлекать друг друга от важных дел, они не могут находиться на одном перекрестке. Лили и Маршалл стоят на соседних перекрестках и хотят попасть на другую пару соседних перекрестков.

Они могут одновременно перейти на соседние перекрестки или кто-то один из них может перейти на соседний перекресток, а другой — остаться на месте. Также они не могут идти по одной и той же улице. В любой момент, когда кто-то из них стоит на перекрестке, другой должен быть на соседнем перекрестке.

Требуется узнать минимальное количество улиц, которое им нужно пройти. Когда они идут по улицам одновременно, обе улицы учитываются. Если они идут по одной и той же улице дважды, она учитывается дважды.

Формат входных данных

В первой строке записаны целые числа n, m, a_1, b_1, a_2, b_2. Здесь n ($3 \leqslant n \leqslant 100$) — количество перекрестков в Нью-Йорке (они занумерованы числами от 1 до n); m ($2 \leqslant m \leqslant 1000$) — количество улиц; a_1, b_1 ($1 \leqslant a_1, b_1 \leqslant n$, $a_1 \neq b_1$) — номера соседних перекрестков, в которых Лили и Маршалл соответственно начинают путь; a_2, b_2 ($1 \leqslant a_2, b_2 \leqslant n$, $a_2 \neq b_2$) — номера соседних перекрестков, куда Лили и Маршалл, соответственно, хотят попасть.($a_1 \neq a_2$ или $b_1 \neq b_2$).

Следующие m строк описывают улицы. В каждой строке записаны два числа p_{i1} и p_{i2} ($1 \leqslant p_{i1}, p_{i2} \leqslant n$, $p_{i1} \neq p_{i2}$) — номера перекрестков, соединенных улицей. Любые два перекрестка соединены не более чем одной улицей.

Формат выходных данных

В первой строке выведите два числа c и k. Здесь c обозначает минимальное количество проходов по улицам; k обозначает количество пар соседних перекрестков, которые Лили и Маршалл посетят во время своего пути, включая a_1, b_1 в начале и a_2, b_2 в конце. Значение k должно быть минимально возможным среди всех планов путешествия с минимальным значением c.

Гарантируется, что решение существует.

Пример

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

,

Задача D. 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 ($1 \leqslant T \leqslant 10$) — число тестов. В первой строке теста заданы два числа N и M ($1 \leqslant N \leqslant 50$, $1 \leqslant M \leqslant 2500$) — число остановок и маршрутов соответственно. Следующие M строк описывают маршруты в формате From To Time ($1 \leqslant$ From, To $\leqslant N$, $1 \leqslant$ Time $\leqslant 100$) — 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 5 5 1 2 1 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	16 -1 42

Задача E. Портал

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

64 мегабайта

Родители подарили Андрюше на Новый Год замечательную компьютерную игру «`Portal 2`». Действие игры происходит на клетчатом поле, в некоторых клетках которого находятся стенки. За один ход игрок может перейти из клетки в любую другую, смежную с ней по стороне. Помимо этого, находясь в любой клетке, можно сделать два выстрела из пушки в двух из четырех направлениях, тогда на месте попадания первого выстрела в стену образуется портал, а на месте попадания второго выстрела — выход из портала. После этого можно войти в портал, и тут же оказаться в выходе.

После использования портал полностью уничтожается. При создании второго портала первый также уничтожается.

На каждом уровне игры требуется добраться из одной клетки поля в другую. На каждый ход уходит ровно одна секунда.

Андрюша очень умный мальчик, и уже выяснил за какое минимальное время можно пройти очередной уровень. А вы сможете?

Формат входных данных

В первой строке через пробел записаны два числа N и M — размеры поля ($4 \leqslant N, M \leqslant 50$).

Следующие N строк содержат по M символов каждая и описывают поле очередного уровня игры. Если j-тый символ i-той строки равен «#», то в ячейке поля с координатами (i, j) находится стенка, иначе ячейка свободна. Начальная позиция игрока обозначена буквой «`S`», клетка, до которой надо добраться — буквой «`T`».

Гарантируется, что можно добраться от начальной клетки до конечной, а также, что игрок стены не дадут выйти игроку за пределы поля.

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Пример

portal.in	portal.out
10 9 ######### #......T# #.....### #.....### #.....### #....S..# #.......# #.......# #.......# #########	3

,

Задача F. Дейкстра

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

Дан ориентированный взвешенный граф.

Найдите кратчайшее расстояние от одной заданной вершины до другой.

Формат входных данных

В первой строке входного файла три числа: N, S и F ($1 \leqslant N \leqslant 2000, 1 \leqslant S, F \leqslant N$), где N — количество вершин графа, S — начальная вершина, а F — конечная. В следующих N строках по N чисел — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое целое неотрицательное число, не превосходящее 10 000 — присутствие ребра данного веса. На главной диагонали матрицы всегда нули.

Формат выходных данных

Вывести искомое расстояние или -1, если пути не существует.

Пример

dijkstra.in	dijkstra.out
3 1 2 0 -1 2 3 0 -1 -1 4 0	6

Задача G. Доставка кефирчика

Имя входного файла:	`kefir.in`
Имя выходного файла:	`kefir.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

64 мегабайта

Во время проведения очередной Межгалактической Летней Компьютерной Школы (МЛКШ) организаторы столкнулись с проблемой доставки кефирчика для вечерки. Дело в том, что кефирчик производят на планете под номером 1, а сами школьники живут на планете n, поэтому на доставку кефирчика тратится довольно большое время, а значит он успевает испортиться.

К счастью, галактическая транспортная система «Берендеев-Экспресс» постепенно внедряет новые кефиропроводы, способные передавать кефир со скоростью, в два раза превышающей скорость старых моделей. А именно, с любой планеты на любую по старым кефиропроводам кефир проходит за два года, а по новым — за один.

Разумеется, грешно было бы не воспользоваться инновационными технологиями, поэтому директор МЛКШ попросил вас написать программу, которая по данным о имеющихся кефиропроводах (как новых, так и старых) узнает кратчайший путь от планеты 1 до планеты n.

Формат входных данных

В первой строке входного файла даны два целых числа n и m ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 100\,000$) — количество планет и количество кефиропроводов соответственно. В последующих m строках даны тройки натуральных чисел u_i, v_i и c_i. Числа u_i и v_i обозначают номера планет, соединенных i-м кефиропроводом, а c_i ($c_i = 1$ или $c_i = 2$) — количество лет, которое потребуется, чтобы передать кефир с одной планеты на другую через i-й кефиропровод. Планеты во входном файле нумеруются с единицы. Кефир по трубопроводам можно передавать в обоих направлениях.

Формат выходных данных

В выходной файл требуется вывести одно число — количество лет, которое требуется, чтобы доставить кефир с планеты 1 на планету n. Если доставка невозможна, то в выходной файл требуется вывести «`-1`».

Примеры

kefir.in	kefir.out
3 2 1 2 2 2 3 1	3
3 1 2 3 1	-1
2 5 1 2 1 1 2 2 1 2 1 1 1 2 2 2 1	1

Задача H. Налоги

Имя входного файла:	`tax.in`
Имя выходного файла:	`tax.out`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Король Байтландии следует мировой тенденции и вводит налоги везде, где только может. Недавно он придумал налог на путешествия, который взымается с каждого, кто путешествует по стране.

Каждой Байтландской дороге сопоставлен размер налога за нее. При прохождении через город путешественнику необходимо заплатить налог, равный максимуму из размеров налога за дорогу, по которой он вошел в город, и дорогу, по которой он вышел из города. Также необходимо заплатить за первый и последний город: для них налог равен единственной соответствующей дороге.

Ваш друг Байтазар собирается в путешествие из Байтгорода в Байтополис. Помогите ему спланировать маршрут таким образом, чтобы размер уплаченного налога был минимален.

Формат входных данных

Первая строка ввода содержит два целых числа n и m ($2 \leqslant n \leqslant 100\,000, 1 \leqslant m \leqslant 200\,000$) — количество городов и количество дорог в Байтландии. Города пронумерованы от 1 до n.

Следующие m строк содержат описания дорог. i-ая из этих строк содержит три целых числа a_i, b_i, c_i ($1 \leqslant a_i, b_i \leqslant n, a_i \neq b_i, 1 \leqslant c_i \leqslant 1\,000\,000$). Это означает, что города a_i и b_i соединены двунаправленной дорогой с размером налога, равным c_i байтлей. Каждая пара соединена не более, чем одной дорогой.

Формат выходных данных

В первой и единственной строке вывода должно содержаться одно целое число — минимальный размер налога (в байтлях) на путешествие из Байтгорода (т.е. города номер 1) в Байтополис (т.е. город номер n). Гарантируется, что существует путь, соединяющий эти два города.

Примеры

tax.in	tax.out
4 5 1 2 5 1 3 2 2 3 1 2 4 4 3 4 8	12
6 6 6 2 4 4 2 3 3 5 3 2 1 8 4 1 2 4 6 6	13

,

Задача I. Кратчайшие пути

Имя входного файла:	`path.in`
Имя выходного файла:	`path.out`
Ограничение по времени:	3.5 секунд
Ограничение по памяти:	256 мегабайт

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 ребёр в графе и номер начальной вершины соответственно ($2 \leqslant n \leqslant 1\,000$, $1 \leqslant m \leqslant 2\,000$).

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 «-».

Пример

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

,

Задача J. Авиаперелеты

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
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 составил набор подходящих авиарейсов, 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: $2 \leqslant N \leqslant 100$, $1 \leqslant M \leqslant 10^5, 1 \leqslant K \leqslant 100, 1 \leqslant S \leqslant N, 1 \leqslant F \leqslant N$.

Далее идет M строк, задающих расписание авиарейсов. i-я строка содержит три натуральных числа: S_i, F_i, 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. $1 \leqslant S_i \leqslant N, 1 \leqslant F_i \leqslant N, 1 \leqslant P_i \leqslant 10^6$.

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.

Пример

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