

Задача А. Путь в графе

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

В неориентированном графе требуется найти длину минимального пути между двумя вершинами.

Формат входных данных

Первым на вход поступает число N — количество вершин в графе ($1 \leq N \leq 100$). Затем записана матрица смежности (0 обозначает отсутствие ребра, 1 — наличие ребра). Далее задаются номера двух вершин — начальной и конечной.

Формат выходных данных

Необходимо вывести длину пути в ребрах. Если пути нет, нужно вывести -1 .

Пример

стандартный ввод	стандартный вывод
5 0 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 3 5	3

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. Рёбро 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. Пифагоров экспресс (для C++)

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

В стране Флатландии есть n городов, расположенных в целочисленных точках плоскости. Транспортная система Флатландии настолько развита, что между любыми двумя городами ходит экспресс имени Пифагора. С помощью него можно добраться от города с координатами x_1, y_1 до города с координатами x_2, y_2 за время $(x_1 - x_2)^2 + (y_1 - y_2)^2$. По каждой линии ходит достаточно поездов, и временем на пересадки можно пренебречь.

Сообщение с внешним миром во Флатландии продумано несколько хуже, и единственный аэропорт международного сообщения находится в городе с номером s . Вам же хочется попасть в город с номером t . Определите, за какое наименьшее время это можно сделать.

Формат входных данных

В первой строке задано число городов n ($1 \leq n \leq 2 \cdot 10^4$). В следующих n строках заданы координаты городов x_i, y_i ($|x_i|, |y_i| \leq 10^4$). В последней строке даны s и t — номера начального и конечного города в пути ($1 \leq s, t \leq n$).

Формат выходных данных

Выведите одно число — минимальное время, за которое можно добраться из s в t .

Пример

стандартный ввод	стандартный вывод
3 0 1 0 0 1 3 2 3	6

Задача D. Флойд

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

Дан ориентированный граф, в котором могут быть кратные ребра и петли. Каждое ребро имеет вес, выражающийся целым числом (возможно, отрицательным). Гарантируется, что циклы отрицательного веса отсутствуют.

Требуется посчитать матрицу длин кратчайших путей от каждой вершины до всех остальных вершин.

Формат входных данных

Программа получает сначала число N ($1 \leq N \leq 100$) — количество вершин графа и число M ($0 \leq M \leq 10^4$) — количество ребер.

В следующих строках идет M троек чисел, описывающих ребра: начало ребра, конец ребра и вес (вес — целое число от -100 до 100).

Формат выходных данных

Требуется вывести матрицу N на N . В j -м столбце i -й строки следует вывести длину кратчайшего пути от вершины i до вершины j , либо число $30\,000$, если не существует пути между данными вершинами.

Пример

стандартный ввод	стандартный вывод
6 4	0 10 20 30000 30000 30000
1 2 10	30000 0 10 30000 30000 30000
2 3 10	30000 30000 0 30000 30000 30000
1 3 100	30000 30000 30000 0 -10 30000
4 5 -10	30000 30000 30000 30000 0 30000
	30000 30000 30000 30000 30000 0

Задача Е. Один голодный конь

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

На шахматной доске $N \times N$ в клетке (x_1, y_1) стоит голодный шахматный конь. Он хочет попасть в клетку (x_2, y_2) , где растет вкусная шахматная трава. Какое наименьшее количество ходов он должен для этого сделать?

Формат входных данных

На вход программы поступает пять чисел: N, x_1, y_1, x_2, y_2 ($5 \leq N \leq 20, 1 \leq x_1, y_1, x_2, y_2 \leq N$).
Левая верхняя клетка доски имеет координаты $(1, 1)$, правая нижняя — (N, N) .

Формат выходных данных

В первой строке выведите единственное число K — наименьшее необходимое число ходов коня.
В каждой из следующих $K + 1$ строк должно быть записано 2 числа — координаты очередной клетки в пути коня.

Пример

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

Задача F. Числа

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

Витя хочет придумать новую игру с числами. В этой игре от игроков требуется преобразовывать четырехзначные числа не содержащие нулей при помощи следующего разрешенного набора действий:

- Можно увеличить первую цифру числа на 1, если она не равна 9.
- Можно уменьшить последнюю цифру на 1, если она не равна 1.
- Можно циклически сдвинуть все цифры на одну вправо.
- Можно циклически сдвинуть все цифры на одну влево.

Например, применяя эти правила к числу 1234 можно получить числа 2234, 1233, 4123 и 2341 соответственно. Точные правила игры Витя пока не придумал, но пока его интересует вопрос, как получить из одного числа другое за минимальное количество операций.

Формат входных данных

Во входном файле содержится два различных четырехзначных числа, каждое из которых не содержит нулей.

Формат выходных данных

Программа должна вывести последовательность четырехзначных чисел, не содержащих нулей. Последовательность должна начинаться первым из данных чисел и заканчиваться вторым из данных чисел, каждое последующее число в последовательности должно быть получено из предыдущего числа применением одного из правил. Количество чисел в последовательности должно быть минимально возможным.

Пример

стандартный ввод	стандартный вывод
1234	1234
4321	2234
	3234
	4323
	4322
	4321

Задача G. Игрушечный лабиринт

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

Игрушечный лабиринт представляет собой прозрачную плоскую прямоугольную коробку, внутри которой есть препятствия и перемещается шарик. Лабиринт можно наклонять влево, вправо, к себе или от себя, после каждого наклона шарик перемещается в заданном направлении до ближайшего препятствия или до стенки лабиринта, после чего останавливается. Целью игры является загнать шарик в одно из специальных отверстий — выходов. Шарик проваливается в отверстие, если оно встречается на его пути (шарик не обязан останавливаться в отверстии).

Первоначально шарик находится в левом верхнем углу лабиринта. Гарантируется, что решение существует и левый верхний угол не занят препятствием или отверстием.

Формат входных данных

В первой строке входного файла записаны числа N и M — размеры лабиринта (целые положительные числа, не превышающие 100). Затем идет N строк по M чисел в каждой — описание лабиринта. Число 0 в описании означает свободное место, число 1 — препятствие, число 2 — отверстие.

Формат выходных данных

Выведите единственное число — минимальное количество наклонов, которые необходимо сделать, чтобы шарик покинул лабиринт через одно из отверстий.

Пример

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

Задача Н. 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.

Пример

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 I. Найти короткий путь

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

Однажды Юрий решил снарядить очередной поход в горы и собрался взять с собой несколько друзей. Горы можно представить как n полян, между которыми есть тропы. Юрий знает тропы двух типов — короткие и длинные. По каждой тропе можно ходить только в одном направлении. Правда, длинные тропы не всегда длиннее коротких.

Для того, чтобы друзья не считали своего проводника скучным человеком, Юрий всегда чередует тропы. Строго говоря, Юрий никогда не ходит по двум тропам одного типа подряд. Юрий рассматривает несколько маршрутов через горы, каждый из которых заканчивается на поляне с номером n , от которой можно добраться до города. Не смотря на длину троп, каждую из них Юрий проходит за одну минуту. Для каждого варианта начальной поляны посчитайте минимальное время прогулки от текущей поляны до поляны с номером n , если Юрий верен своим принципам и никогда не ходит по двум тропам одного типа подряд.

Формат входных данных

В первой строке через пробел заданы два числа n и m — количество полян и троп соответственно ($2 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$).

В следующих m строках записаны три числа: a_i, b_i и $type_i$, означающие, что с поляны a_i на поляну b_i ведет тропа типа $type_i$, где $type_i \in \{1, 2\}$ ($type_i$ означает тип тропы).

Обратите внимание, что тропа может соединять поляну саму с собой, а также между двумя полянами может быть несколько троп.

Формат выходных данных

Выведите через пробел $n - 1$ число — минимальное время прогулки от вершины с номером i до вершины с номером n . Если от какой-то вершины нельзя добраться до вершины с номером n , выполняя принципы Юрия, выведите вместо минимального времени число -1 .

Пример

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

Задача J. Полнейший Флойд

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

Вам дан **ориентированный** граф. Если в нём есть отрицательные циклы, то выведите хотя бы один из них. В противном случае выведите матрицу попарных кратчайших расстояний между вершинами.

Формат входных данных

На первой строке даны $1 \leq V \leq 500$ — число вершин в графе и $0 \leq E \leq \min(2.5 \times 10^4, |V| \times |V|)$ — число рёбер в графе. Далее записаны рёбра графа, по одному в строчке. Каждое ребро записано в виде трёх целых чисел: $1 \leq from, to \leq |V|$, $|w| \leq 10^5$.

Формат выходных данных

На первой строк выведите LOOP или NO LOOP в зависимости от наличия отрицательного цикла.

Далее, если в графе есть отрицательный цикл, выведите на второй строке его длину, а на третьей строке выведите его вершины в любом порядке, соблюдая направление рёбер графа. В противном случае выведите V строк по V чисел, где j -й элемент на i -й строке означает кратчайшее расстояние между вершинами i и j и INF означает, что пути между вершинами нет.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 1 2 3 2 1 3 2	NO LOOP 0 1 2 INF 0 2 INF INF 0
3 3 1 2 -1 2 3 -1 3 1 -1	LOOP 3 3 1 2
7 7 2 1 -4 2 7 -17 3 1 -14 4 5 16 4 5 -19 4 5 14 7 6 -15	NO LOOP 0 INF INF INF INF INF INF -4 0 INF INF INF -32 -17 -14 INF 0 INF INF INF INF INF INF INF 0 -19 INF INF INF INF INF INF 0 INF INF INF INF INF INF INF 0 INF INF INF INF INF INF -15 0

Задача К. Заправки

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. В каждом городе бак бензина имеет разную стоимость. Вам требуется добраться из первого города в N -ый, потратив как можно меньше денег. Покупать бензин впрок нельзя.

Формат входных данных

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

Примеры

стандартный ввод	стандартный вывод
5 3 6 1 7 6 8 1 2 5 4 5 1 3 4 5 2 2 4 2 3 3 1	3
5 3 7 2 9 4 4 1 2 1 3 2 3 4 5	-1

Задача L. Корованы

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

Внимание: слово «корован» не содержит орфографических ошибок, так как является интернет-мемом (если не верите, можете после контеста посмотреть здесь: <http://lurkmore.to/Корованы>). Однако во время сдачи ЕГЭ по русскому языку советуем всё же использовать вариант написания «караван».

Студент Илья часто прогуливает пары в университете. Друзья осуждают его за это, но они не знают, что всё освободившееся время Илья проводит не за просмотром сериалов или слушая музыку. Он делает компьютерную игру своей мечты! Действие игры происходит в лесу. Там есть эльфы, деревянные домики и злой. А ещё там можно грабить корованы... И хоть в данный момент в игре только один корован, у Ильи уже возникли трудности с реализацией процесса ограбления.

Игровой мир можно представить в виде нескольких поселений, соединённых дорогами. От любого поселения до любого другого можно добраться по дорогам (возможно, заходя в другие поселения по пути). Все поселения пронумерованы целыми числами от 1 до n . Все дороги двусторонние и имеют одинаковую длину, равную единице. Перемещаться вне дорог нельзя. Корован идёт из поселения s в поселение f по одному из кратчайших путей. Поселение r является логовом злого. Днём группа разбойников из поселения r получила задание ограбить корован. Вечером этого же дня они раздобудут точный план пути, по которому корован пойдёт на следующее утро. В течение ночи разбойники могут переместиться в любое поселение, которое будет посещено корованом, даже в поселения s и f , и, устроив там засаду, утром ограбить корован. Разумеется, среди всех таких поселений разбойники выберут ближайшее к поселению r . До вечера ещё далеко, и пока они не знают, по какому пути пойдёт корован, но хотят знать, какое максимальное расстояние им придётся преодолеть в худшем случае до поселения, в котором они смогут его ограбить. Помогите Илье посчитать это расстояние, и тогда, возможно, он снова начнёт ходить на пары!

Формат входных данных

В первой строке даны целые числа n и m ($3 \leq n \leq 10^5$; $2 \leq m \leq 10^5$) — количество поселений в игровом мире и количество дорог между ними.

В следующих m строках дано описание дорог. В каждой из них записаны целые числа a и b — номера поселений, между которыми есть дорога. Гарантируется, что любая дорога соединяет два различных поселения. Также гарантируется, что сеть дорог является связной.

В последней строке записаны целые числа s , f и r — номера поселений, описанных в условии. Гарантируется, что эти поселения попарно различны.

Формат выходных данных

В единственной строке выведите искомое расстояние.

Пример

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

Замечание

В примере корован может пойти либо по пути $1 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 7$, либо по пути $1 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 7$.

В первом случае разбойники устроят засаду в поселении 4, которое находится на расстоянии 1 от логова злого, во втором случае — в поселении 2 или в поселении 6, которые находятся на расстоянии 2 от логова злого.

Второй вариант хуже для разбойников, поэтому его и нужно выдать в качестве ответа.

Задача М. Телепорты

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

Маленький Эдсгер живёт на прямой в точке S , и хочет попасть к своему другу Роберту в точку T . На прямой есть n телепортов, i -й телепорт находится в точке x_i и может мгновенно перенести на расстояние ровно d_i (то есть в точки $x_i - d_i$ и $x_i + d_i$). Кроме использования телепортов, можно ходить по прямой пешком со скоростью 1. За какое минимальное время можно дойти из S в T ?

Обратите внимание, что Эдсгер не обязан использовать телепорт, если находится в одной с ним точке.

Формат входных данных

В первой строке записаны три целых числа n , S и T ($0 \leq n \leq 10^5$, $0 \leq S, T \leq 10^9$).

В следующих n строках описаны телепорты по одному в строке. Описание каждого телепорта состоит из двух целых чисел x_i, d_i ($0 \leq x_i \leq 10^9$, $1 \leq d_i \leq 10^9$).

Гарантируется, что все телепорты находятся в разных точках.

Формат выходных данных

Выведите одно число — минимальное время, необходимое для того, чтобы добраться из S в T .

Примеры

стандартный ввод	стандартный вывод
0 10 20	10
1 10 20 11 8	2
1 20 10 11 8	10
1 10 20 9 9	3
1 10 20 15 1000	10
2 10 20 10 1000 1009 988	2

Задача N. Дед и мопед

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Дед Максим собирается в путешествие по Флатландии. К сожалению, из доступных средств передвижения у него есть только мопед, запас хода которого ограничен. Более точно, если бак мопеда полностью заполнен, то мопед может проехать не более s километров без дополнительной дозаправки.

Всего во Флатландии есть n городов, пронумерованных от 1 до n . В некоторых городах находятся заправки, и если в городе есть заправка, то в этом городе дед Максим может полностью наполнить бак. К сожалению, заправки присутствуют лишь в k городах. Также во Флатландии есть m дорог, i -я из которых соединяет города u_i и v_i и имеет длину c_i километров. По каждой дороге можно перемещаться в обоих направлениях.

Дед Максим начинает свое путешествие в городе с номером 1 с полным баком (в городе 1 есть заправка). Помогите ему определить, до каких городов он сможет добраться на мопеде.

Формат входных данных

В первой строке записаны четыре целых числа n, m, k, s ($1 \leq n \leq 150\,000$, $0 \leq m \leq 150\,000$, $1 \leq k \leq n$, $1 \leq s \leq 10^9$) — количество городов, количество дорог, количество заправок и объем бака.

В следующих m строках записаны по три целых числа u_i, v_i, c_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq c_i \leq s$) — начало, конец и длина i -й дороги. Гарантируется, что не существует двух дорог, соединяющих одинаковую пару городов.

В следующей строке записаны k целых чисел p_i ($1 \leq p_i \leq n$) — номера городов с заправками. Гарантируется, что заправка присутствует в городе 1.

Формат выходных данных

В первой строке выведите единственное число x : количество гордов, до которых дед Максим может добраться.

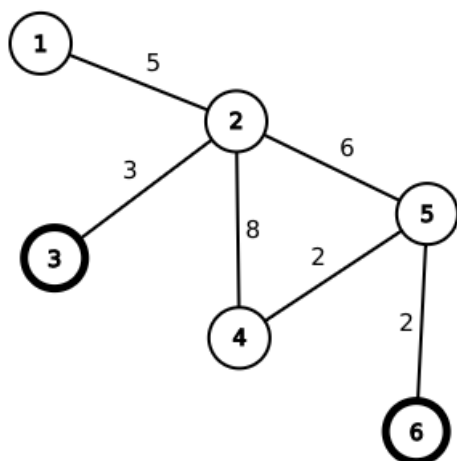
Во второй строке выведите x целых чисел — номера подходящих городов **в порядке возрастания**.

Пример

стандартный ввод	стандартный вывод
6 6 3 10 1 2 5 2 3 3 2 4 8 2 5 6 4 5 2 5 6 2 1 3 6	4 1 2 3 5

Замечание

Рисунок ниже иллюстрирует первый пример из условия:



Из начального города с номером 1 можно доехать до городов с номерами 2 и 3 без дополнительных дозаправок. Также можно доехать до города с номером 3, пополнить там бак и доехать до города с номером 5. До городов 4 и 6 добраться невозможно, так как минимальное расстояние от достижимой заправки до них равно 11.

Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		k	n	c_i		
0	0	—	—	—	—	Тесты из условия.
1	15	—	$n \leq 100$	—	0	
2	11	$k = 1$	$n \leq 5000$	—	—	
3	8	$k = 1$	—	—	2	
4	12	—	$n \leq 5000$	$c_i = 1$	—	
5	9	—	$n \leq 5000$	$c_i \leq 10$	0, 4	
6	17	—	$n \leq 5000$	—	0, 1, 2, 4, 5	
7	28	—	—	—	0–6	

Задача O. Roadblock

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

Каждое утро Фермер Джон идет от дома к амбару. Ферма представляет собой множество из n полей (дом на поле 1, амбар на поле n), соединенных m двусторонними дорогами, с каждой из которых ассоциирована длина.

Никакие два поля не соединены более чем одной дорогой, и существует маршрут дорог от любого поля к любому. Когда ФД идет от одного поля к другому, он всегда выбирает маршрут, состоящий из последовательности дорог, которые дают минимальную суммарную длину.

Коровы решили сделать ФД маленькую неприятность, выложив сено на одной из m дорог, тем самым удваивая ее длину.

Коровы хотят выбрать такую дорожку, чтобы максимально увеличить расстояние, которое ФД пройдет от дома до амбара. Помогите коровам определить, насколько они удлинят маршрут ФД.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n \leq 300$; $1 \leq m \leq 90\,000$).

Следующие m строк описывают дороги. i -я из этих строк состоит из трех целых чисел u_i , v_i и l_i — дорога длины l_i соединяет поля с номерами v_i и u_i ($1 \leq u_i, v_i \leq n$; $1 \leq l_i \leq 10^6$).

Формат выходных данных

Выведите максимально возможное увеличение общей длины кратчайшего маршрута, которого можно добиться удвоением длины одной дороги.

Примеры

стандартный ввод	стандартный вывод
5 7 2 1 5 1 3 1 3 2 8 3 5 7 3 4 3 2 4 7 4 5 2	2

Замечание

Если коровы удвоят длину дороги от поля 3 к полю 4 (от 3 до 6), тогда кратчайшим маршрутом станет путь 1-3-5, с общей длиной $1 + 7 = 8$. Что на 2 больше, чем исходный кратчайший маршрут.

Задача Р. 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 Q. Налоги

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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 взимается 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 ($2 \leq n \leq 100\,000, 1 \leq m \leq 200\,000$) — 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 ($1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq c_i \leq 1\,000\,000$). 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

стандартный ввод	стандартный вывод
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 R. Липецкие дороги

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

Вася приехал на Вселипецкую олимпиаду школьников по информатике в город L. В городе есть n перекрестков, пронумерованных числами от 1 до n , некоторые из которых соединены двунаправленными дорогами. В городе L есть почти все: воздушное метро, аэропорт и, конечно, такси.

Всего в городе есть две фирмы такси. Чтобы заказать такси, необходимо открыть приложение фирмы и выбрать две точки — номера перекрестков, откуда и куда нужно добраться. Для каждой фирмы известно, по каким дорогам может перемещаться такси этой фирмы, а также за какое время оно может проехать каждую дорогу. Во всех такси есть навигатор, который прокладывает самый быстрый маршрут в точку назначения по доступным дорогам.

Все в этом городе хорошо, но интернет не везде работает хорошо. Поэтому заказать себе такси можно только на определенных перекрестках. Время на загрузку приложений различное для всех перекрестков. В самом отеле есть интернет, и там можно мгновенно выбрать приложение и заказать такси.

Вася проспал, и теперь опаздывает на первый тур. Помогите ему как можно быстрее добраться на такси от отеля до места проведения олимпиады. Так как Вася хочет добраться до места проведения олимпиады максимально быстро, он хочет действовать оптимально, поэтому иногда он может выходить из такси на некотором перекрестке, после чего вызывать новое такси и продолжать поездку на нем.

Отель и место проведения олимпиады находятся на разных перекрестках.

Формат входных данных

В первой строке записаны четыре целых числа: n , m_1 , m_2 и k ($2 \leq n \leq 100\,000$, $0 \leq m_1 + m_2 \leq 200\,000$, $0 \leq k \leq n$) — количество перекрестков, дорог для первой фирмы, дорог для второй фирмы и перекрестков с работающим интернетом соответственно.

Далее в m_1 строках перечислены дороги, по которым может перемещаться такси первой фирмы по одной в строке. Каждая из дорог задается тремя числами a_i , b_i , w_i ($1 \leq a_i, b_i \leq n$, $0 \leq w_i \leq 10^6$), где a_i , b_i — номера перекрестков, между которыми есть дорога, а w_i — время, за которое такси первой фирмы может преодолеть эту дорогу.

Далее в m_2 строках перечислены дороги для второй фирмы в аналогичном формате.

Далее в k строках перечислены номера домов ind_i , около которых работает интернет, и время загрузки приложений ($0 \leq c_i \leq 10^6$).

В последней строке находятся два числа s , f ($s \neq f$) — адреса отеля и места проведения олимпиады.

Между любой парой перекрестков существует не более одной дороги для каждого такси. Дорога не может соединять перекресток с самим собой.

Не гарантируется, что из любого перекрестка можно добраться до любого другого.

Формат выходных данных

Выведите единственное целое число — наименьшее возможное время прибытия к месту проведения олимпиады. Если же до места нельзя добраться, выведите -1.

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	подзадача	—
1	10	$2 \leq n \leq 5$	подзадача	—
2	10	$2 \leq n \leq 300$	подзадача	1
3	10	$2 \leq n \leq 1000, c_i = 0, k = n$	подзадача	—
4	10	$c_i = 0, k = n$	подзадача	3
5	20	$c_i = 1, w_i = 1$	подзадача	—
6	20	$2 \leq n \leq 1000, 0 \leq m_1 + m_2 \leq 20\,000$	подзадача	1, 2
7	20	Дополнительных ограничений нет	подзадача	1, 2, 3, 4, 5, 6

Примеры

стандартный ввод	стандартный вывод
7 6 6 2 1 2 10 1 3 1 2 3 1 2 4 3 3 5 5 4 6 3 1 5 10 2 3 2 3 4 5 4 5 10 4 7 30 6 7 100 2 6 5 1 1 7	45
3 1 1 0 1 2 1 2 3 1 1 3	-1

Замечание

В первом примере можно действовать так: воспользоваться такси первой фирмы и проехать маршрут $1 \rightarrow 3 \rightarrow 2$ за 2 единицы времени, затем выйти, вызвать за 6 единиц времени такси второй фирмы и проехать маршрут $2 \rightarrow 3 \rightarrow 4 \rightarrow 7$ за 37 единиц времени.

Во втором примере ни на одном перекрестке не работает интернет, а из отеля нельзя доехать до точки проведения олимпиады при помощи такси какой-либо одной фирмы.

Задача S. Скандинавские контрабандисты

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

После того, как ограничения на приграничную торговлю между Норвегией и Швецией были сняты, многие предприниматели стали пользоваться возможностью торговать своими товарами с соседней страной. Однако, власти обеспокоены тем, что отсутствие ограничений поспособствует росту контрабанды товаров на границе. Для предотвращения данной угрозы таможенные службы Норвегии и Швеции объединились в борьбе с общим врагом.

Для того, чтобы успешно пройти таможню, необходимо посетить некоторое количество КПП. Всего на границе есть n КПП, пронумерованных числами от 1 до n , где КПП под номером 1 является входом на таможню, а КПП с номером n — выходом уже по другую сторону границы. Также были построены m дорог с двусторонним движением, которые соединяют некоторые пары КПП. Для прохождения i -го КПП требуется t_i минут.

Каждый КПП может контролироваться либо норвежской, либо шведской таможенной службой, причем Норвегия должна контролировать k КПП, а Швеция, соответственно, $n - k$ КПП. Известно, что если некоторая дорога соединяет два КПП, контролируемых таможенной службой одной и той же страны, то все контрабандисты, перемещающиеся по данной дороге, обязательно будут пойманы, чего нельзя сказать об остальных дорогах. Также известно, что контрабандисты всегда спешат и хотят пройти таможню как можно быстрее, поэтому они всегда выбирают кратчайший по затратам времени путь из КПП с номером 1 в КПП с номером n .

От вас требуется распределить все КПП между таможенными службами Норвегии и Швеции таким образом, чтобы любой контрабандист, направляющийся из КПП с номером 1 в КПП с номером n , был пойман.

Формат входных данных

В первой строке записаны три целых числа n , m и k ($2 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$, $0 \leq k \leq n$) — количество КПП, количество дорог, а также количество КПП, которые нужно отдать под управление норвежской таможенной службы, соответственно.

Во второй строке через пробел записаны n положительных чисел $t_1, t_2, \dots, t_n$ ($1 \leq t_i \leq 10^4$) — количество времени, необходимое для прохождения каждого КПП.

В каждой из следующих m строк записаны два целых числа u и v ($1 \leq u, v \leq n$) — это означает, что между КПП с номерами u и v имеется дорога с двусторонним движением.

Гарантируется, что от любого КПП можно добраться до любого другого КПП, пользуясь только имеющимися дорогами. Между каждой парой КПП может быть не более одной дороги. Никакая дорога не соединяет КПП с самим собой.

Формат выходных данных

Если существует способ распределить все КПП между таможенными службами Норвегии и Швеции требуемым образом, выведите строку, состоящую из n символов «N» и «S», где символ s_i равен «N», если i -й КПП следует отдать Норвегии, и «S» — в противном случае.

Если требуемого распределения КПП не существует, выведите строку «impossible» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
3 2 0 1 1 1 1 2 2 3	SSS
2 1 1 1 1 1 2	impossible
8 9 4 3 3 1 2 2 3 2 1 1 2 1 3 1 4 2 5 3 6 4 7 5 8 6 8 7 8	NNNNSSSS

Задача Т. Пляж

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

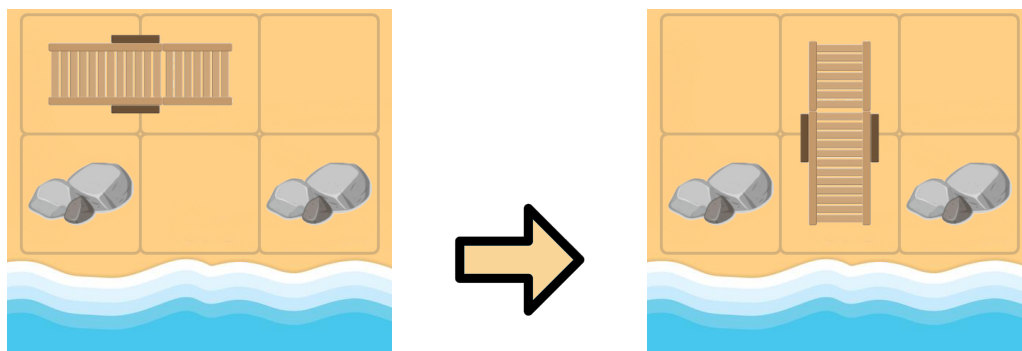
Андрей обожает море. Именно поэтому в разгар летнего сезона он решил отправиться на пляж, взяв с собой лежак, чтобы позагорать.

Пляж представляет собой прямоугольное поле, состоящее из n строк и m столбцов. Некоторые клетки этого поля свободны, на некоторых расположены дорожки, камни, ларьки и другие несдвигаемые объекты, а так же на двух соседних по стороне клетках могут стоять лежаки, расположенные как горизонтально, так и вертикально.

Андрей надеется поставить где-то свой лежак, но вот незадача, свободных мест для него может уже не быть! Именно поэтому Андрей просит вас помочь найти ему свободное место для лежака. Лежак Андрея тоже должен занимать две соседние по стороне клетки.

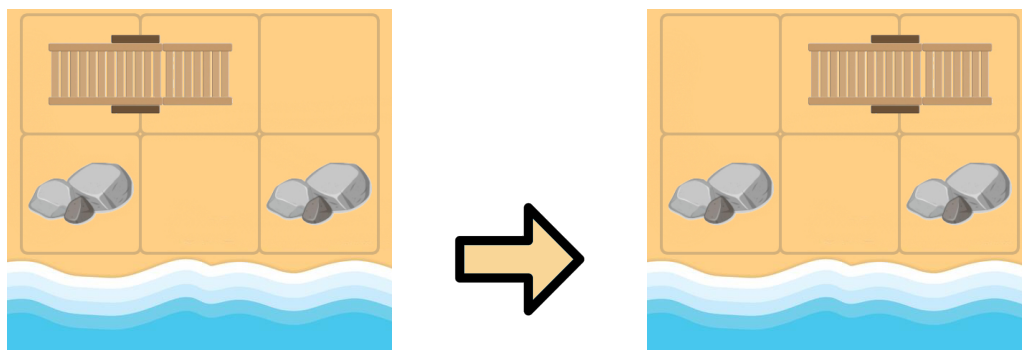
Если двух соседних свободных клеток нет, то для того, чтобы освободить место для лежака, придется потревожить других туристов. Для этого вы можете делать следующие действия:

- Подойти к любому лежаку, и доставив p единиц дискомфорта его хозяину, поднять лежак за одну из сторон и повернуть на 90 градусов. Одна из половин лежака должна остаться в той же клетке, а вторая переместиться в свободную клетку. При этом во время поворота на пути лежака может быть что угодно.



Поворот лежака на 90 градусов вокруг клетки (1, 2).

- Подойти к любому лежаку, и доставив q единиц дискомфорта его хозяину, подвинуть лежак вдоль его длинной стороны на одну клетку. Одна сторона лежака должна переместиться на место другой, а другая — в свободную клетку.



Сдвиг лежака на одну клетку вправо.

В любой момент времени каждый лежак занимает две соседние клетки, не содержащие ничего другого. Вы можете двигать одновременно максимум один лежак.

Помогите Андрею освободить место для еще одного лежака, доставив минимальное количество дискомфорта окружающим, или скажите, что сделать это не получится.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 300\,000$, $1 \leq n \cdot m \leq 300\,000$) — количество строк и столбцов на поле.

Вторая строка содержит два целых числа p и q ($1 \leq p, q \leq 10^9$) — количество единиц дискомфорта, которые приносят поворот и сдвиг лежака, соответственно.

Каждая из следующих n строк содержит m символов, описывающие клетки поля. Все строки состоят из символов «L», «R», «D», «U», «.» и «#», задающих тип клетки. Символы «L», «R», «D» и «U» обозначают половину лежака, находящегося в этой клетке — левая, правая, нижняя или верхняя половина, соответственно. Символ «.» обозначает свободную клетку, а символ «#» — клетку, занятую несдвигаемым объектом.

Формат выходных данных

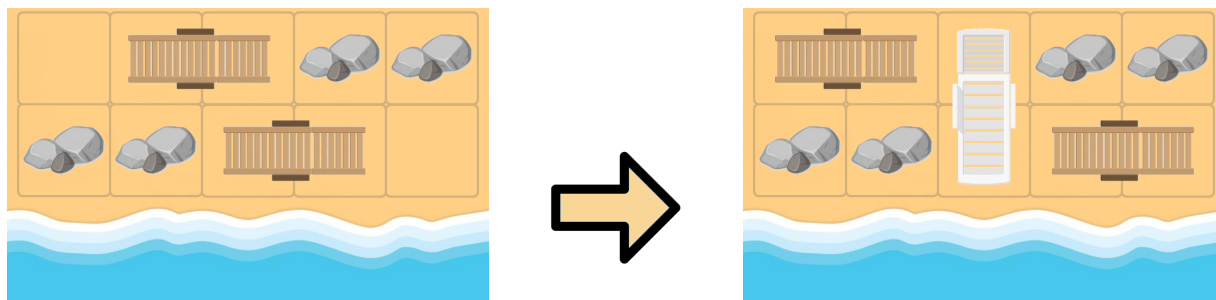
Выведите одно число — минимальное количество дискомфорта, которое возникнет при освобождении места для лежака. Если освободить место для лежака невозможно, выведите число -1 .

Примеры

стандартный ввод	стандартный вывод
2 5 5 2 .LR## ##LR.	4
2 3 4 5 LR. #.#	-1
4 3 10 10 .LR ### UU# DD.	-1
3 6 10 7 .U##. # #DLR## .##LR.	24

Замечание

В первом примере, передвинув верхний лежак налево, а нижний лежак направо, Андрей сможет вертикально поставить лежак посередине пляжа. Таким образом, мы доставим $2 + 2 = 4$ единицы дискомфорта. Можно показать, что доставить меньшее количество дискомфорта не получится.



Оптимальные действия в первом примере (лежака Андрея обозначен белым цветом).

Во втором примере из условия невозможно освободить место для лежака Андрея. Всевозможные состояния пляжа после поворотов и сдвигов показаны в условии задачи.

Задача U. Доставка пиццы

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

Алиса живет в New Tsukuba City и учится в колледже. Завтра в городе начнется новый социальный эксперимент, связанный со сменой направлений движения на односторонних дорогах города. Каждый день рано утром направление одной из дорог будет сменено на противоположное, после чего поздно вечером того же дня направление будет изменено на начальное.

Более формально: всего в городе имеется m дорог, пронумерованных числами от 1 до m . В i -й день утром направление i -й дороги будет изменено на противоположное, а вечером направление i -й дороги станет исходным.

Каждый день Алиса заказывает пиццу из одной и той же пиццерии. Разумеется, пицца доставляется по кратчайшему маршруту от пиццерии до дома Алисы.

За счет смены направлений дорог кратчайший путь может меняться изо дня в день. Помогите Алисе понять, как проведение эксперимента повлияет на время доставки пиццы.

Формат входных данных

В первой строке записаны два числа n и m ($2 \leq n \leq 10^5, 1 \leq m \leq 10^5$) — количество перекрестков в городе и количество односторонних дорог соответственно.

Каждая из следующих m строк содержит три числа a_i, b_i и c_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq c_i \leq 10^5$). Эти числа означают, что i -я дорога соединяет перекрестки с номерами a_i и b_i , причем можно проехать по ней в направлении $a_i \rightarrow b_i$ (и только в этом направлении) за время c_i .

Обратите внимание, что несколько различных дорог могут соединять два перекрестка.

Пиццерия находится на перекрестке с номером 1, а дом Алисы — на перекрестке с номером 2.

Гарантируется, что до начала эксперимента от пиццерии до дома Алисы существует маршрут.

Формат выходных данных

Выведите m строк. В i -й строке выведите:

- HAPPY, если кратчайший путь в i -й день станет короче;
- SOSO, если длина кратчайшего пути в i -й день не изменится;
- SAD, если в i -й день кратчайший путь станет длиннее, либо в i -й день не будет пути от пиццерии до дома Алисы.

Алисе не важно, сможет ли доставщик пиццы добраться от ее дома до пиццерии на обратном пути.

Примеры

стандартный ввод	стандартный вывод
4 5 1 3 5 3 4 6 4 2 7 2 1 18 2 3 12	SAD SAD SAD SOSO HAPPY HAPPY
7 5 1 3 2 1 6 3 4 2 4 6 2 5 7 5 6	SOSO SAD SOSO SAD SOSO SOSO
10 14 1 7 9 1 8 3 2 8 4 2 6 11 3 7 8 3 4 4 3 2 1 3 2 7 4 8 4 5 6 11 5 8 12 6 10 6 7 10 8 8 3 6	SOSO SAD HAPPY SOSO SOSO SOSO SAD SOSO SOSO SOSO SOSO SOSO SOSO SAD SAD

Задача V. Два безопасных выхода

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

Вам дан простой связный неориентированный граф G с N вершинами и M рёбрами.

Вершины графа пронумерованы от 1 до N , рёбра — от 1 до M . Рёбро с номером i соединяет вершины U_i и V_i . Между двумя вершинами, соединёнными ребром, можно передвигаться в обе стороны, причём переход по одному ребру занимает одну единицу времени.

Каждая вершина является либо *безопасной*, либо *опасной*. Эта информация задана строкой S длины N , состоящей из символов 'S' и 'D'. Если i -й символ строки S равен 'S', то вершина i считается безопасной, если же он равен 'D', то вершина i считается опасной.

Гарантируется, что в графе есть как минимум две безопасные вершины и как минимум одна опасная вершина.

Для каждой опасной вершины v нужно найти следующее значение:

Минимальное возможное время, за которое можно начать движение из некоторой безопасной вершины, пройти через вершину v и оказаться в безопасной вершине, отличной от начальной.

Путь может проходить через любые вершины и рёбра графа, в том числе повторять их, но начальная и конечная вершины должны быть безопасными и различными.

Ответ нужно вывести для всех опасных вершин.

Формат входных данных

Первая строка содержит два целых числа N и M ($3 \leq N \leq 2 \cdot 10^5$, $N - 1 \leq M \leq 2 \cdot 10^5$) — число вершин и рёбер графа.

Следующие M строк содержат описания рёбер. В i -й из этих строк записаны два целых числа U_i и V_i ($1 \leq U_i, V_i \leq N$, $U_i \neq V_i$) — концы i -го ребра.

Гарантируется, что между каждой парой вершин не более одного ребра, и граф является связным.

Последняя строка содержит строку S длины N , состоящую только из символов 'S' и 'D'.

Гарантируется, что в S не менее двух символов 'S' и не менее одного символа 'D'.

Формат выходных данных

Пусть в графе K опасных вершин. Выведите K строк: на i -й строке должно быть выведено искомое значение для i -й опасной вершины, если все опасные вершины упорядочить по номеру по возрастанию.

Примеры

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

Задача W. Отражения

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

Дана прямоугольная таблица из H строк и W столбцов. Клетка в i -й строке сверху и j -м столбце слева обозначается как (i, j) .

В каждой клетке расположен тип зеркала, обозначаемый символом 'A', 'B' или 'C':

- тип 'A' — в клетке нет зеркала, луч просто проходит клетку насквозь;
- тип 'B' — в клетке стоит зеркало по диагонали из левого верхнего угла в правый нижний (символически это выглядит как символ '\');
- тип 'C' — в клетке стоит зеркало по диагонали из правого верхнего угла в левый нижний (символически как '/').

Рассматриваются четыре направления: вверх, вниз, влево и вправо. Лучи отражаются по следующим правилам:

- В клетке типа 'A' направление не меняется.
- В клетке типа 'B' (диагональ "\"):
 - если луч идёт вправо, после выхода из клетки он идёт вверх;
 - если луч идёт вверх, после выхода он идёт вправо;
 - если луч идёт влево, после выхода он идёт вниз;
 - если луч идёт вниз, после выхода он идёт влево.
- В клетке типа 'C' (диагональ "/"):
 - если луч идёт вправо, после выхода из клетки он идёт вниз;
 - если луч идёт вниз, после выхода он идёт вправо;
 - если луч идёт влево, после выхода он идёт вверх;
 - если луч идёт вверх, после выхода он идёт влево.

Takahashi стоит слева от клетки $(1, 1)$ и светит фонариком строго вправо, то есть луч входит в клетку $(1, 1)$ слева и сначала направлен вправо. Aoki стоит справа от клетки (H, W) .

Лучи не рассеиваются и всегда движутся по одной прямой, пока не упрутся в границу таблицы. При выходе за пределы таблицы процесс заканчивается. Если луч выходит из таблицы через правую сторону клетки (H, W) , считаем, что он достиг Aoki.

Вы можете выполнять следующую операцию любое (в том числе нулевое) количество раз:

- выбрать клетку (i, j) и заменить её тип на любой другой (то есть 'A' заменить на 'B' или 'C', 'B' — на 'A' или 'C', 'C' — на 'A' или 'B').

Требуется найти минимальное количество операций, достаточное для того, чтобы существовал путь луча из начального положения до Aoki, то есть чтобы луч, начавшись слева от клетки $(1, 1)$ и двигаясь по описанным правилам, вышел через правую сторону клетки (H, W) .

Вам нужно ответить на несколько независимых тестов.

Формат входных данных

Первая строка содержит одно целое число T ($1 \leq T$) — число тестов.

Далее следуют описания тестов.

Описание каждого теста состоит из следующих строк:

- первая строка содержит два целых числа H и W ($1 \leq H, W, H \cdot W \leq 2 \cdot 10^5$) — размеры таблицы;
- далее следуют H строк по одной, i -я строка содержит строку S_i длины W , состоящую из символов 'A', 'B', 'C'. Символ $S_i[j]$ задаёт тип клетки (i, j) .

Гарантируется, что сумма $H \cdot W$ по всем тестам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого теста выведите одно целое число — минимальное количество операций, которое необходимо выполнить, чтобы луч можно было довести до Aoki. Если луч уже можно довести без операций, выведите 0.

Пример

стандартный ввод	стандартный вывод
4	0
3 4	2
ABCB	10
CACC	5
BCBA	
2 2	
CB	
AA	
1 10	
BCBСBСBСBС	
10 10	
ССААВАСААА	
СССВАСАССА	
ВАСААВСВВА	
АСССААСССА	
ССААААССВА	
ААСВВАССАА	
ВСССАСВВАВ	
СВВСААСССС	
СВВССВСВСА	
ВВАСАВВАСС	

Задача X. 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 ($1 \leq n, m \leq 2000$) — количество строк и столбцов в лабиринте, соответственно.

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

Третья строка содержит два целых числа x, y ($0 \leq x, y \leq 10^9$) — 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.

Примеры

стандартный ввод	стандартный вывод
4 5 3 2 1 2***. ...** *.....	10
5 5 5 4 3 1 **... **.*. ...*. .***.	16

Задача Y. Метро

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

Метрополитен состоит из нескольких линий метро. Все станции метро в городе пронумерованы натуральными числами от 1 до N . На каждой линии расположено несколько станций. Если одна и та же станция расположена сразу на нескольких линиях, то она является станцией пересадки и на этой станции можно пересест с любой линии, которая через нее проходит, на любую другую (опять же проходящую через нее).

Напишите программу, которая по данному вам описанию метрополитена определит, с каким минимальным числом пересадок можно добраться со станции A на станцию B . Если данный метрополитен не соединяет все линии в одну систему, то может так получиться, что со станции A на станцию B добраться невозможно, в этом случае ваша программа должна это определить.

Формат входных данных

Сначала вводится число N — количество станций метро в городе ($2 \leq N \leq 100$). Далее следует число M — количество линий метро ($1 \leq M \leq 20$). Далее идет описание M линий. Описание каждой линии состоит из числа P_i — количество станций на этой линии ($2 \leq P_i \leq 50$) и P_i чисел, задающих номера станций, через которые проходит линия (ни через какую станцию линия не проходит дважды).

Затем вводятся два различных числа: A — номер начальной станции, и B — номер станции, на которую нам нужно попасть. При этом если через станцию A проходит несколько линий, то мы можем спуститься на любую из них. Так же если через станцию B проходит несколько линий, то нам не важно, по какой линии мы приедем.

Формат выходных данных

Выведите минимальное количество пересадок, которое нам понадобится. Если добраться со станции A на станцию B невозможно, программа должна вывести одно число -1 .

Пример

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