

## Задача А. Экскурсия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Группа из  $n$  человек решила поехать на экскурсию. В процессе экскурсии можно заехать в некоторые из  $m$  городов.

Экскурсовод попросила каждого человека высказать свои пожелания по поводу посещения городов. Каждый человек может про какие-то города заявить, что он хочет их непременно посетить, а про какие-то — что точно не хочет.

Группа всегда путешествует вместе. Если группа заезжает в город, то все люди, заявившие, что точно не хотят его посетить, расстраиваются. Если группа не заезжает в город, то расстраиваются все люди, которые заявили, что хотят его непременно посетить.

Экскурсовод понимает, что удовлетворить все пожелания не всегда можно. Она хочет составить план, чтобы каждый человек расстроился не более одного раза.

Помогите экскурсоводу справиться с этой нелегкой задачей и составьте план посещения городов или выясните, что это невозможно.

### Формат входных данных

В первой строке входных данных содержатся три целых числа  $n$ ,  $m$ ,  $k$  — количество человек, количество городов и количество пожеланий ( $1 \leq n, m, k \leq 100\,000$ ).

В каждой из последующих  $k$  строк содержатся по два целых числа  $a$  и  $b$ , обозначающих пожелания ( $1 \leq a \leq n, 1 \leq |b| \leq m$ ). Если  $b > 0$ , то человек под номером  $a$  хочет посетить город под номером  $b$ . Если же  $b < 0$ , то человек под номером  $a$  не хочет посетить город под номером  $-b$ . Каждое пожелание указано во вводе не более одного раза, ни для кого из участников нет одновременно пожелания посетить и не посещать один и тот же город.

### Формат выходных данных

Если решения не существует, то выведите  $-1$ .

Иначе, в первой строке выходных данных выведите единственное целое число  $k$  — количество городов, которые войдут в план.

Во второй строке выведите  $k$  целых чисел — номера городов, которые следует посетить. Номера городов можно выводить в любом порядке.

Если существует несколько возможных ответов, можно вывести любой из них. Обратите внимание, что не требуется искать максимальное или минимальное  $k$ , можно вывести любой корректный ответ.

### Примеры

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

## Задача В. Магнитные подушки

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

Город будущего застроен небоскребами, для передвижения между которыми и парковки транспорта многие тройки небоскребов соединены треугольной подушкой из однополярных магнитов. Каждая подушка соединяет ровно 3 небоскреба и вид сверху на нее представляет собой треугольник, с вершинами в небоскребах. Это позволяет беспрепятственно передвигаться между соответствующими небоскребами. Подушки можно делать на разных уровнях, поэтому один небоскреб может быть соединен различными подушками с парами других, причем два небоскреба могут соединять несколько подушек (как с разными третьими небоскребами, так и с одинаковым). Например, возможны две подушки на разных уровнях между небоскребами 1, 2 и 3, и, кроме того, магнитная подушка между 1, 2, 5.

Система магнитных подушек организована так, что с их помощью можно добраться от одного небоскреба, до любого другого в этом городе (с одной подушки на другую можно перемещаться внутри небоскреба), но поддержание каждой из них требует больших затрат энергии.

Требуется написать программу, которая определит, какие из магнитных подушек нельзя удалять из подушечной системы города, так как удаление даже только этой подушки может привести к тому, что найдутся небоскребы из которых теперь нельзя добраться до некоторых других небоскребов, и жителям станет очень грустно.

### Формат входных данных

В первой строке входного файла находятся числа  $N$  и  $M$  — количество небоскребов в городе и количество работающих магнитных подушек соответственно ( $3 \leq N \leq 100000$ ,  $1 \leq M \leq 100000$ ). В каждой из следующих  $M$  строк через пробел записаны три числа — номера небоскребов, соединенных подушкой. Небоскребы пронумерованы от 1 до  $N$ . Гарантируется, что имеющиеся магнитные подушки позволяют перемещаться от одного небоскреба до любого другого.

### Формат выходных данных

Выведите в выходной файл сначала количество тех магнитных подушек, отключение которых невозможно без нарушения сообщения в городе, а потом их номера. Нумерация должна соответствовать тому порядку, в котором подушки перечислены во входном файле. Нумерация начинается с единицы.

### Примеры

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

## Задача С. Минимизация мостов

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

Добавить в граф  $G = \langle V, E \rangle$  (возможно несвязный, с петлями и кратными рёбрами) ровно одно ребро, так чтобы количество мостов в данном графе стало минимально возможным.

Напомним, что мостом в графе называется такое ребро, удаление которого увеличивает число компонент связности графа.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  – количества вершин и рёбер графа соответственно ( $1 \leq n \leq 200\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $v_i$ ,  $u_i$  – номерами концов ребра ( $1 \leq v_i, u_i \leq n$ ).

### Формат выходных данных

Выведите наименьшее число мостов, которое можно получить добавлением ровно одного ребра.

### Пример

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

## Задача D. Размещение данных

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

Телекоммуникационная сеть крупной IT-компании содержит  $n$  серверов, пронумерованных от 1 до  $n$ . Некоторые пары серверов соединены двусторонними каналами связи, всего в сети  $m$  каналов. Гарантируется, что сеть серверов устроена таким образом, что по каналам связи можно передавать данные с любого сервера на любой другой сервер, возможно с использованием одного или нескольких промежуточных серверов.

Множество серверов  $A$  называется отказоустойчивым, если при недоступности любого канала связи выполнено следующее условие. Для любого не входящего в это множество сервера  $X$  существует способ передать данные по остальным каналам на сервер  $X$  хотя бы от одного сервера из множества  $A$ .

На рис. 1 показан пример сети и отказоустойчивого множества из серверов с номерами 1 и 4. Данные на сервер 2 можно передать следующим образом. При недоступности канала между серверами 1 и 2 — с сервера 4, при недоступности канала между серверами 2 и 3 — с сервера 1. На серверы 3 и 5 при недоступности любого канала связи можно по другим каналам передать данные с сервера 4.

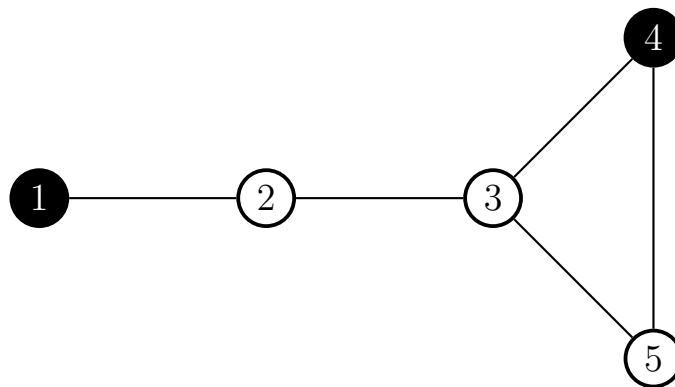


Рис. 1: Пример сети и отказоустойчивого множества серверов.

В рамках проекта группе разработчиков компании необходимо разместить свои данные в сети. Для повышения доступности данных и устойчивости к авариям разработчики хотят продублировать свои данные, разместив их одновременно на нескольких серверах, образующих отказоустойчивое множество. Чтобы минимизировать издержки, необходимо выбрать минимальное по количеству серверов отказоустойчивое множество. Кроме того, чтобы узнать, насколько гибко устроена сеть, необходимо подсчитать количество способов выбора такого множества, и поскольку это количество способов может быть большим, необходимо найти остаток от деления этого количества способов на число  $10^9 + 7$ .

Требуется написать программу, которая по заданному описанию сети определяет следующие числа:  $k$  — минимальное количество серверов в отказоустойчивом множестве серверов,  $s$  — остаток от деления количества способов выбора отказоустойчивого множества из  $k$  серверов на число  $10^9 + 7$

### Формат входных данных

Первая строка входного файла содержит целые числа  $n$  и  $m$  — количество серверов и количество каналов связи соответственно ( $2 \leq n \leq 200\,000$ ,  $1 \leq m \leq 200\,000$ ). Следующие  $m$  строк содержат по два целых числа и описывают каналы связи между серверами. Каждый канал связи задается двумя целыми числами: номерами серверов, которые он соединяет.

Гарантируется, что любые два сервера соединены напрямую не более чем одним каналом связи, никакой канал не соединяет сервер сам с собой, и для любой пары серверов существует способ

передачи данных с одного из них на другой, возможно с использованием одного или нескольких промежуточных серверов.

### Формат выходных данных

Выведите два целых числа, разделенных пробелом:  $k$  — минимальное число серверов в отказоустойчивом множестве серверов,  $c$  — количество способов выбора отказоустойчивого множества из  $k$  серверов, взятое по модулю  $10^9 + 7$ .

### Пример

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

## Задача E. 2-SAT

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

Формулировка 2-SAT: нужно подобрать значения  $n$  булевых переменных так, чтобы все  $m$  утверждений вида  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  обратились в истину. В данной задаче вам гарантируется, что решение существует.

### Формат входных данных

Входной файл состоит из одного или нескольких тестов.

Каждый тест описывается следующим образом. На первой строке число переменных  $n$  и число утверждений  $m$ . Каждая из следующих  $m$  строк содержит числа  $i_1, e_1, i_2, e_2$ , задает утверждение  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  ( $0 \leq i_j < n$ ,  $0 \leq e_j \leq 1$ ). Ограничения: сумма всех  $n$  не больше 100 000, сумма всех  $m$  не больше 300 000.

### Формат выходных данных

Для каждого теста выведите строку из  $n$  нулей и единиц — значения переменных. Если у данной задачи 2-SAT есть несколько решений, выведите любое.

### Пример

стандартный ввод	стандартный вывод
1 0	0
2 2	01
0 0 1 0	000
0 1 1 1	
3 4	
0 1 1 0	
0 0 2 1	
1 1 2 0	
0 0 0 1	

## Задача F. Интересная экскурсия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Во Флатландии  $n$  городов, соединенных  $m$  односторонними дорогами.

Туристическая компания планирует разработать живописный циклический маршрут по дорогам Флатландии. Маршрут должен начинаться и заканчиваться в одном и том же городе и проходить по дорогам в их направлении, посещая промежуточные города. Маршрут может посещать один город несколько раз, но должен проходить по каждой дороге не более одного раза.

Каждая дорога характеризуется типом своего пейзажа, который задается числом от 1 до  $m$ . Чтобы туристический маршрут был живописным, любые две соседние дороги в этом маршруте должны иметь разный тип пейзажа. Это же требование относится к первой и последней дороге маршрута, чтобы можно было начинать путешествовать, начиная с любого города маршрута.

Помогите компании разработать удовлетворяющий этим критериям маршрут, либо выясните, что сделать это невозможно.

### Формат входных данных

Входные данные состоят из нескольких тестов. В первой строке находится число  $T$  — количество тестов ( $1 \leq T \leq 10^5$ ).

Первая строка описания каждого теста содержит два целых числа  $n$  и  $m$  — количество городов и дорог ( $2 \leq n, m \leq 2 \cdot 10^5$ ). Следующие  $m$  строк содержат по три целых числа и описывают дороги в формате  $u_i v_i c_i$  — город, из которого выходит  $i$ -я дорога, город, в который она ведет, и тип её пейзажа ( $1 \leq u_i, v_i \leq n$ ;  $1 \leq c_i \leq m$ ;  $u_i \neq v_i$ ).

Сумма  $n$  по всем тестам не превосходит  $2 \cdot 10^5$ . Сумма  $m$  по всем тестам не превосходит  $2 \cdot 10^5$ .

### Формат выходных данных

Выведите ответ для каждого теста.

Если искомого маршрута не существует, следует вывести число «-1». Иначе, выведите число  $k$  — длину маршрута ( $2 \leq k \leq m$ ). В следующей строке выведите  $k$  чисел  $e_1, e_2, \dots, e_k$  — номера дорог в том порядке, в каком они идут в этом маршруте. Все номера  $e_i$  должны быть различны. Если подходящих маршрутов несколько, можно вывести любой из них.

**Пример**

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



## Задача G. Мастерство тракториста

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Вредный культурист Ишма Дровкин обожает срывать учебные процессы. Особенно он любит заставлять школьников проявлять физическую активность. Вот сейчас, проявив недюжинный ораторский талант, он убедил школьников угнать трактор с хутора и отправиться кататься на нём по полю. Однако, трактор, который им предстоит угнать, старше половины преподавательского состава параллели В, вместе взятой, поэтому им невозможно рулить. Он может ехать вперёд и только вперёд.

Более формально — представим поле в форме прямоугольника  $H \times W$ . Некоторые клетки этого поля заблокированы и по ним невозможно проехать даже на тракторе. Трактор может быть повернут вдоль направления движения по вертикали, горизонтали или диагонали.

Школьники в ЛКШ — люди не только упорные, но и развитые физически, поэтому их не смущает перспектива разворачивать трактор своими руками каждый раз, когда им требуется повернуть. Но, естественно, это не самое приятное времяпрепровождение, так что они просят вас найти маршрут, следуя по которому они будут вынуждены работать руками минимальное количество раз. Важная деталь заключается в том, что сейчас трактор валяется на поле в перевернутом состоянии, так что по меньшей мере один раз им придётся повернуть трактор в правильное положение, которое они могут выбрать удобным им образом. То, как именно будет повернут трактор в конечной точке, не имеет значения.

### Формат входных данных

В первой строке входного файла заданы целые числа  $H, W$  ( $1 \leq H, W \leq 1000$ ) — высота и ширина поля.

В последующих  $H$  строках задано поле, заблокированные клетки обозначены латинскими буквами "X", а свободные — точками.

После поля следует строка с двумя целыми числами  $sx, sy$  ( $1 \leq sx \leq W, 1 \leq sy \leq H$ ) — координаты стартового положения трактора.

Последней строкой идут два целых числа  $tx, ty$  ( $1 \leq tx \leq W, 1 \leq ty \leq H$ ) — координаты конечного положения трактора.

Координаты отсчитываются от нижнего левого угла поля. Стартовое положение не совпадает с конечным. Стартовая и конечная клетки не являются заблокированными.

### Формат выходных данных

Если Ишма Дровкин настолько большой шутник, что пути между стартовой и конечной клетками не существует, выведите  $-1$ . Иначе выведите единственное натуральное число — минимальное количество поворотов, которые школьники должны будут сделать по пути к конечной клетке.

### Пример

стандартный ввод	стандартный вывод
5 7 XX...X X.XXX. ..XXX.X X.X...X ...XXX 1 1 6 5	3

### Замечание

В примере школьники сначала перевернут трактор и развернут его направо. Потом они проедут три клетки и повернут трактор на 45 градусов против часовой стрелки, чтобы тот смотрел по диаго-

нали вверх и вправо. Затем они проедут две клетки и снова повернут трактор на 45 градусов против часовой стрелки, чтобы тот смотрел вверх. Наконец, они проедут три клетки вверх и окажутся в конечной клетке.

## Задача Н. Странствующий торговец

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

Вы прибыли в Австралию, где есть  $n$  рынков, соединённых  $m$  односторонними дорогами, путешествие по каждой дороге занимает определённое количество минут.

На рынках торгуются  $k$  предметами. Каждый предмет имеет определённую стоимость покупки или продажи. Бывает так, что на рынке можно только купить товар или только продать товар, а также бывает, что рынку вообще не интересен товар. Вы можете считать, что если на рынке есть товар, его есть бесконечно много, а также, если рынок готов покупать товар, он готов покупать бесконечно много.

Чтобы как можно быстрее заработать денег вы хотите найти самый эффективный цикл. Цикл — это путь, который начинается в каком-то рынке  $v$  с пустым рюкзаком, проходит по дорогам и рынкам (возможно, по пути покупаются и продаются товары), и возвращается в вершину  $v$ , опять с пустым рюкзаком. Цикл может посещать дорогу или рынок несколько раз. Когда вы покупаете товар, вы кладёте его в рюкзак. Однако в рюкзак можно положить **не более одного товара**. Вы можете считать, что независимо от того, сколько у вас денег, вы можете купить товар.

Выгода цикла — это суммарное количество денег, которое вы заработали на продажах, минус количество денег, которые вы потратили на покупку. Длительность цикла — количество минут, которые вы потратите, чтобы пройти его. Эффективность цикла — отношение его выгоды к длительности.

Найдите максимальную эффективность среди всех циклов со строго положительной длительностью. Вы должны найти это значение, округленное вниз. Если такого цикла не существует, ответ равен 0.

### Формат входных данных

Первая строка содержит три целых числа  $n$ ,  $m$ ,  $k$  ( $1 \leq n \leq 100$ ,  $1 \leq m \leq 9900$ ,  $1 \leq k \leq 1000$ ).

Затем следуют  $n$  строк,  $i$ -я из которых содержит  $2k$  чисел  $b_{i,1}, s_{i,1}, b_{i,2}, s_{i,2}, \dots, b_{i,k}, s_{i,k}$  ( $0 < s_{i,j} \leq b_{i,j} \leq 10^9$ ). Для всех  $1 \leq j \leq k$  пара чисел  $b_{i,j}$  и  $s_{i,j}$  означает цену, по которой вы можете купить и продать товар  $j$  на  $i$ -м рынке, соответственно. Если товар не может быть куплен или продан, тогда значение равно  $-1$ .

Далее следуют  $m$  строк,  $p$ -я из которых содержит три целых числа  $v_p$ ,  $w_p$  и  $t_p$  ( $v_p \neq w_p$ ,  $1 \leq t_p \leq 10^7$ ), описывающих дорогу из  $v_p$  в  $w_p$ , которая занимает  $t_p$  минут.

Гарантируется, что не существует такой пары рёбер  $1 \leq p < q \leq m$ , что  $(v_p, w_p) = (v_q, w_q)$ .

### Формат выходных данных

Выведите одно число — ответ.

### Пример

стандартный ввод	стандартный вывод
4 5 2 10 9 5 2 6 4 20 15 9 7 10 9 -1 -1 16 11 1 2 3 2 3 3 1 4 1 4 3 1 3 1 1	2

## Задача I. Проездной

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

ЮИ-кун живёт в городе с  $n$  станциями, занумерованными от 1 до  $n$ . Есть  $m$  железных дорог, пронумерованных от 1 до  $m$ , и  $i$ -я дорога соединяет станции  $a_i$  и  $b_i$  в обоих направлениях, притом стоимость проезда равна  $c_i$ .

ЮИ-кун живёт рядом со станцией  $s$  и ходит в старшую школу ЮИ рядом со станцией  $t$ . Он планирует купить проездной между этими станциями. Когда он покупает проездной, он должен выбрать какой-то кратчайший путь между  $s$  и  $t$ . Используя проездной, он может использовать любую железную дорогу на выбранном пути в обоих направлениях, не платя за проезд.

ЮИ-кун часто ездит в книжные магазины рядом со станциями  $u$  и  $v$ . Поэтому он хочет купить проездной так, чтобы минимальная стоимость проезда от  $u$  до  $v$  была минимальна.

Когда он перемещается из станции  $u$  на станцию  $v$ , он сперва выбирает путь от станции  $u$  до станции  $v$ . Тогда за дороги в пути, входящие в проездной, он заплатит 0 иен, а за не входящие в проездной — стоимость проезда через них, то есть  $c_i$  для дороги  $i$ .

Найдите минимально возможную стоимость пути из  $u$  в  $v$ , если ЮИ-кун выберет путь для проездного оптимально.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $2 \leq n \leq 10^5$ ,  $1 \leq m \leq 2 \cdot 10^5$ ).

Вторая строка содержит два целых числа  $s$  и  $t$  ( $1 \leq s, t \leq n$ ,  $s \neq t$ ).

Третья строка содержит два целых числа  $u$  и  $v$  ( $1 \leq u, v \leq n$ ,  $u \neq v$ ,  $s \neq u$  или  $t \neq v$ ).

Далее следуют  $m$  строк, описывающие железные дороги,  $i$ -я из них содержит три целых числа  $a_i$ ,  $b_i$ ,  $c_i$  ( $1 \leq a_i < b_i \leq n$ ,  $1 \leq c_i \leq 10^9$ ). Для всех  $1 \leq i < j \leq m$  верно  $a_i \neq a_j$  или  $b_i \neq b_j$ .

Гарантируется, что ЮИ-кун может добраться от любой станции до любой другой, перемещаясь только по железным дорогам.

### Формат выходных данных

Выведите одно число — ответ.

## Примеры

стандартный ввод	стандартный вывод
6 6 1 6 1 4 1 2 1 2 3 1 3 5 1 2 4 3 4 5 2 5 6 1	2
6 5 1 2 3 6 1 2 1000000000 2 3 1000000000 3 4 1000000000 4 5 1000000000 5 6 1000000000	3000000000
8 8 5 7 6 8 1 2 2 2 3 3 3 4 4 1 4 1 1 5 5 2 6 6 3 7 7 4 8 8	15
5 5 1 5 2 3 1 2 1 2 3 10 2 4 10 3 5 10 4 5 10	0

## Задача J. Диаграммы Юнга выходят в интернет

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

Рома и Сеня дали констест из 10 задач про диаграмму Юнга и теперь вынуждены скрываться от ДемидКомНадзора. Но они слишком любят диаграммы Юнга и хотят делиться новыми открытиями. Однако держаться вместе очень рискованно, так как в случае чего повяжут их обоих, поэтому они вынуждены общаться через Интернет. Но "обычный Интернет" полностью контролируется ДемидКомНадзором, поэтому они пользуются даркнетом.

В даркнете любое сообщение может проделать длинный запутанный путь до получателя через множество серверов. Более того, оно даже может проходить через один и тот же сервер несколько раз. За счет этого сообщение сложнее отследить.

Компьютер Ромы связан с сервером 1, а компьютер Сени — с сервером  $n$ .

ДемидКомНадзор хочет перехватить Ромино сообщение с диаграммой Юнга. Для этого ему необходимо взломать такой сервер, что сообщение, посланное Ромой, по любому пути к Сене пройдет через этот сервер **ровно один** раз.

Найдите все подходящие сервера.

### Формат входных данных

В первой строке файла дано количество тестовых примеров  $t$  ( $1 \leq t \leq 500$ ).

Каждый тестовый пример выглядит так: в первой строке даны два числа:  $n$  и  $m$  ( $2 \leq n \leq 5 \cdot 10^5$ ,  $0 \leq m \leq 10^6$ ), число серверов и число прямых соединений между серверами.

В каждой из последующих  $m$  строк содержится упорядоченная пара чисел  $a$  и  $b$  ( $1 \leq a, b \leq n$ ), это означает, что с сервера  $a$  можно переслать сообщение напрямую на сервер  $b$ .

Гарантируется, что эти упорядоченные пары не повторяются внутри одного тестового примера.

Так же гарантируется, что и сумма по  $n$ , и сумма по  $m$  по всем тестовым примерам не превосходит  $10^6$ .

### Формат выходных данных

Для каждого тестового примера требуется вывести две строки: в первой число подходящих серверов, а во второй — номера этих серверов в порядке их следования на пути от  $a$  до  $b$  через пробел.

### Пример

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

## Задача К. Красно-синий граф

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

Вам дан граф на  $n$  вершинах с  $\frac{n \cdot (n-1)}{2}$  ориентированными ребрами. Ребро между вершинами  $(u, v)$  существует, если  $u < v$ . Каждое ребро покрашено либо в синий, либо в красный цвет. Проверьте, существует ли такая пара вершин  $(v, u)$ , что существует:

- путь  $v \rightarrow u$ , состоящий только из красных ребер
- путь  $v \rightarrow u$ , состоящий только из синих ребер

### Формат входных данных

В первой строке вводится число  $n$  ( $1 \leq n \leq 5000$ ). В следующих  $n - 1$  строках вводится описание графа.

$i$ -я строка описания графа будет содержать строку из  $n - i$  символов 'R' или 'B'.  $j$ -й символ обозначает цвет ребра  $i \rightarrow (i + j)$ .

### Формат выходных данных

Если искомая пара  $v, u$  существует, выведите «YES». Иначе выведите «NO»

### Примеры

стандартный ввод	стандартный вывод
3 RB R	YES
3 RR R	NO

## Задача L. 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 М. 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.

### Пример

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