

Задача А. Два пути

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

В Байтландии города соединены односторонними дорогами. Дорожная система Байтландии обладает одним интересным свойством: если вы выехали из города по какой-то дороге, вы не сможете вернуться назад. Иными словами, структура дорог может быть описана ориентированным ациклическим графом.

Такая особенность создает некоторые проблемы. Например, до некоторых городов нельзя добраться даже из столицы Байтландии. Бывает и хуже: дороги часто закрываются на ремонт. В таком случае до некоторых городов нельзя будет добраться, даже если до них можно было доехать ранее.

Байтазер живет в столице и часто путешествует в другие города Байтландии. Для каждого города C он хотел бы знать, существуют ли два пути из столицы в C , которые не имеют общих дорог. Если это правда (или C является столицей), Байтазер знает, что путешествие до C всегда возможно, даже если какая-то из дорог закрыта на ремонт. Помогите Байтазеру найти все города, до которых можно добраться из столицы, даже если какая-нибудь одна дорога закроется на ремонт.

Формат входных данных

Первая строка содержит два числа n, m ($1 \leq n \leq 200\,000$, $1 \leq m \leq 500\,000$) – количество городов и дорог в Байтландии. Города пронумерованы числами $1, 2, \dots, n$, столица имеет номер 1. Следующие m строк описывают дорог: i -я из них содержит два числа u_i, v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) означающие, что i -я односторонняя дорога начинается в городе u_i и заканчивается в городе v_i .

Формат выходных данных

В первой строке выведите количество городов, до которых можно добраться до столицы, даже если одна из дорог закроется на ремонт. В следующей строке выведите номера этих городов по возрастанию, разделенных пробелами.

Пример

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

Задача В. Полезные дороги

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

В столице Берляндии n перекрестков и m дорог, соединяющих их. По дорогам можно передвигаться лишь в одном направлении. Как вы догадываетесь, в Берляндии две беды: дураки и дороги.

До выборов мэра остался один месяц, поэтому сейчас самое время для текущего правительства делать город лучше. Чтобы показать, что они заботятся об инфраструктуре и бюджете, правительство решило починить только *полезные* дороги.

Текущий мэр считает дорогу от перекрестка u к перекрестку v полезной, если существует простой путь, содержащий дорогу (u, v) , начинающийся в мэрии и заканчивающийся в каком-либо перекрестке. Путь называется простым, если никакой перекресток на этом пути не повторяется. Мэрия находится на перекрестке номер 1.

Помогите министерству транспорта найти все полезные дороги в городе.

Формат входных данных

Вход содержит несколько тестовых примеров. Каждый тест начинается со строки, содержащей два целых числа n и m ($2 \leq n \leq 2 \cdot 10^5; 1 \leq m \leq 2 \cdot 10^5$) — количество перекрестков и дорог в городе. Следующие m строк описывают дороги. Описание состоит из пары чисел u_i и v_i ($1 \leq u_i, v_i \leq n; u_i \neq v_i$), означающих, что i -я дорога начинается в перекрестке u_i и заканчивается в перекрестке v_i . Перекрестки пронумерованы от 1 до n . Мэрия находится на перекрестке номер 1.

Гарантированно, что между каждой парой перекрестков не более одной дороги в каждом направлении.

Каждый тестовый пример завершается пустой строкой. Сумма n по всем тестовым примерам не превосходит $2 \cdot 10^5$. Аналогично для m : сумма m по всем тестовым примерам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера выведите две строки. В первой строке должно находиться количество полезных дорог. Во второй строке находятся индексы полезных дорог в возрастающем порядке. Дороги пронумерованы от 1 до m в том же порядке, как они идут во входном файле. Если полезных дорог нет, оставьте вторую строку пустой.

Пример

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

Задача С. Дерево доминаторов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	512 мегабайт

Дан ориентированный граф с n вершинами и m ребрами. Пусть S - это вершина 1. Вершина A доминирует над вершиной B , если любой путь из S в B проходит через A . В частности, вершина S доминирует над всеми вершинами. *Непосредственный доминатор* вершины B (далее $idom(v)$) - это самый близкий доминатор к вершине B на произвольно выбранном пути из S в B . Далее непосредственный доминатор обозначается за $idom(B)$.

Формат входных данных

Вход содержит несколько тестовых примеров. Каждый тест начинается со строки, содержащей два целых числа n и m ($2 \leq n \leq 2 \cdot 10^5$; $1 \leq m \leq 2 \cdot 10^5$) — количество вершин и ребер в графе. Следующие m строк описывают ребра. Описание состоит из пары чисел u_i и v_i ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$), означающих, что i -е ребро начинается в вершину u_i и заканчивается в вершине v_i . Вершины пронумерованы от 1 до n .

Гарантировано, что между каждой парой вершин не более одного ребра в каждом направлении.

Каждый тестовый пример завершается пустой строкой. Сумма n по всем тестовым примерам не превосходит $2 \cdot 10^5$. Аналогично для m : сумма m по всем тестовым примерам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера выведите n чисел в одну строку через пробел. i -е число в строке должно равняться $idom(i)$. Если вершина i не достижима из вершины 1, считайте $idom(i)$ равным -1. Также $idom(1)$ считайте равным 1.

Пример

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

Задача D. Увеличение стоимости

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

Берляндия состоит из n городов, пронумерованных от 1 до n . Город номер 1 является столицей Берляндии. Также в Берляндии m двунаправленных дорог между некоторыми городами. Разные дороги пересекаются только в городах. Между каждой парой городов не больше одной дороги, и никакая дорога не соединяет город с собой. Если вы передвигаетесь по j -й дороге в любом направлении, вы платите пошлину c_j . **По заданным m дорогам можно добраться до каждого города от столицы.**

Вы руководите компанией, занимающейся доставкой, ваш главный офис находится в столице. Ваша компания доставляет различные посылки в каждый город Берляндии, поэтому для каждого города вы выбрали путь, минимизирующий суммарную пошлину всех дорог в этом пути. Обозначим суммарную пошлину на пути до вершины k как d_k .

Государство решило выбрать **ровно одну** дорогу (вы не знаете, какую именно) и увеличить пошлину за ее использование. Для каждой дороги вы хотели бы знать, сколько городов будут затронуты этим увеличением. Город k считается затронутым изменением, если после увеличения пошлины за использование дороги, вы не сможете выбрать путь суммарной стоимости d_k .

Формат входных данных

Первая строка содержит два числа n и m : количество городов и дорог в Берляндии ($2 \leq n \leq 2 \cdot 10^5$, $n - 1 \leq m \leq 2 \cdot 10^5$).

Каждая из следующих m дорог содержит по три целых числа: u_j , v_j , и c_j ($1 \leq u_j, v_j \leq n$, $1 \leq c_j \leq 10^9$), означающих, что j -я дорога соединяет города u_j и v_j , а пошлина за ее использование равна c_j .

Между каждой парой городов не более одной дороги, никакая дорога не соединяет город с собой. Гарантируется, что из столицы можно добраться до любого города по заданным дорогам.

Формат выходных данных

Выведите m чисел, каждое в отдельной строке. Число в строке j должно быть равно количеству городов, затронутых увеличением пошлины j -й дороги.

Пример

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

Задача Е. Максимизация суммы

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

Дан массив a_1, a_2, \dots, a_n . За одну операцию можно выбрать число x ($1 \leq x \leq n$) и для всех $i = x, 2x, 3x, \dots, x \cdot \lfloor \frac{n}{x} \rfloor$ заменить a_i на 0. Необходимо определить максимальную сумму массива после выполнения любого количества таких операций.

Формат входных данных

Первая строка содержит единственное целое число n ($1 \leq n \leq 100$) — размер массива.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — элементы массива.

Формат выходных данных

Выведите максимальную сумму массива, которую можно получить, выполнив любое количество подобных операций.

Примеры

стандартный ввод	стандартный вывод
6 1 2 -6 4 5 3	12
6 100 -100 -100 -100 100 -100	200
5 -1 -2 -3 -4 -5	0
2 -1000 100000	99000

Задача F. Два языка

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

Родители Эллиота говорят с ним дома по-английски и по-французски. Он слышал множество разных слов, но не всегда уверен, какие слова принадлежат какому языку. Эллиот знает одно предложение, про которое точно уверен, что оно на английском, и знает одно предложение, про которое уверен, что оно на французском. И еще несколько предложений, которые могут быть или на английском, или на французском.

Если слово встречается в английском предложении, оно должно быть словом в английском языке. Если слово встречается во французском предложении, оно должно быть словом во французском языке.

Вам даны все предложения, которые слышал Эллиот. Найдите минимальное возможное количество слов, среди тех, которые он слышал, которые должны принадлежать сразу обоим языкам.

Формат входных данных

В первой строке содержится количество тестов T ($1 \leq T \leq 25$). Далее следует T тестов. Каждый начинается со строки, содержащей целое число N ($1 \leq N \leq 200$). Далее следует N строк, каждая содержит последовательность слов, разделенных пробелом. Каждое слово состоит только из строчных латинских букв. Первая из этих N строк — предложение на английском. Вторая — предложение на французском. Остальные могут быть как предложениями на английском, так и предложениями на французском.

Каждое слово содержит не более 10 символов. Первые два предложения содержат не более 1 000 слов каждое. Остальные содержат не более чем по 10 слов.

Формат выходных данных

Для каждого теста выведите в новой строке одно число — ответ на задачу.

Пример

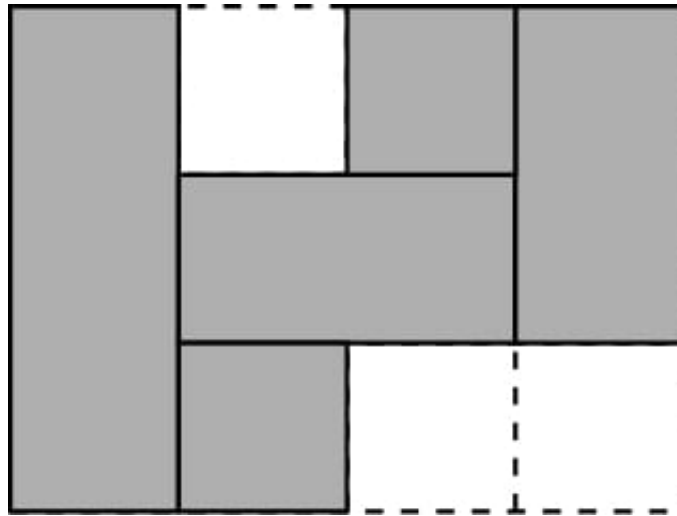
стандартный ввод
4 2 he loves to eat baguettes il aime manger des baguettes 4 a b c d e f g h i j a b c i j f g h d e 4 he drove into a cul de sac elle a conduit sa voiture il a conduit dans un cul de sac il mange pendant que il conduit sa voiture 6 adieu joie de vivre je ne regrette rien adieu joie de vivre je ne regrette rien a b c d e f g h i j a b c i j f g h d e
стандартный вывод
1 4 3 8

Задача G. Кирпичи

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Кирпич — прямоугольник с целыми сторонами шириной 1 или высотой 1 (или и то и другое).

Дана сетка $n \times m$, и каждая ячейка окрашена в черный или белый цвет. Замощение — это способ поместить кирпичи на сетку так, чтобы каждая черная ячейка была покрыта **ровно одним** кирпичом, а каждая белая ячейка не была покрыта кирпичом. Другими словами, кирпичи размещаются только в черных ячейках, покрывают все черные ячейки, и **никакие два кирпича не перекрываются**.



Пример замощения с первого примера с использованием 5 кирпичей. Существует также замощение из 4 кирпичей.

Какое минимальное количество кирпичей необходимо для замощения данной сетки?

Формат входных данных

Первая строка содержит два целых числа n, m ($1 \leq n, m \leq 200$) — количество строк и столбцов соответственно.

Следующие n строки описывают сетку. i -я строка содержит строку длиной m , где j -я строка обозначает цвет ячейки в строке i , столбец j . Черная ячейка обозначается символом «#», а белая — символом «.».

Гарантируется, что есть хотя бы одна черная ячейка.

Формат выходных данных

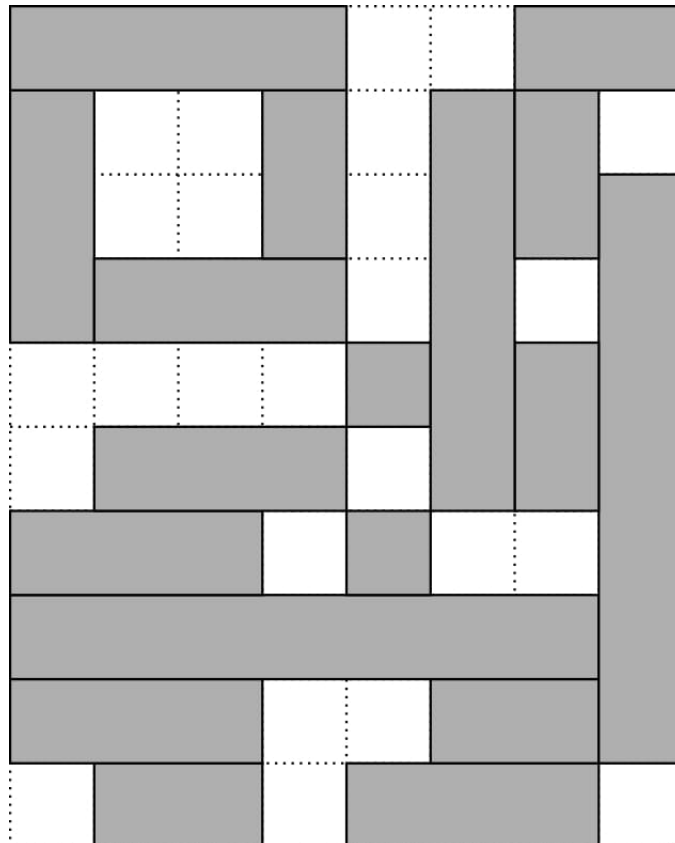
Выведите единственное целое число — минимальное количество требуемых кирпичей.

Примеры

стандартный ввод	стандартный вывод
<pre>3 4 #.# ### ##..</pre>	4
<pre>6 6 ##### ##... ##### ##...# ##...# #####</pre>	6
<pre>10 8 ####.## #.#.##. #.#.### ####.#.# ...#### .###.### ###.#.# ##### ###.### .##.###.</pre>	18

Замечание

Сетка с первого примера может быть замощена 4-мя кирпичами, размещенными вертикально.
 Сетка с третьего примера может быть замощена такими 18 кирпичами:



Задача Н. П(л)отно

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

Определим плотность непустого неориентированного графа, как $\frac{\text{количество рёбер}}{\text{количество вершин}}$.

Даны два числа N и D , а так же неориентированный граф G из N вершин и DN рёбер. Вершины пронумерованы от 1 до N , i -е ребро соединяет вершины v_i и u_i .

Требуется проверить, удовлетворяет ли данный граф следующему свойству:

- Для любого непустого подмножества вершин X графа G плотность графа, индуцированного на вершинах X , **строго** меньше, чем D .

Формат входных данных

Первая строка содержит единственное число T ($1 \leq T$) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа N и D ($1 \leq N, D$) — количество вершин, и параметр D .

Следующие DN строк содержат по два целых числа v_i и u_i ($1 \leq v_i < u_i \leq n$) — концы i -го ребра.

Гарантируется, что в графе нет кратных рёбер.

Гарантируется, что $\sum DN \leq 50\,000$.

Формат выходных данных

Выведите «Yes», если граф удовлетворяет условию, и «No» иначе.

Пример

стандартный ввод	стандартный вывод
2	Yes
3 1	No
1 2	
1 3	
2 3	
4 1	
1 2	
1 3	
2 3	
3 4	