Задача A. Проблема падишаха

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

Мудрый падишах внимательно следит за благополучием своих подданных, когда вершит их судьбы. В частности, на нем все заботы о вступающих в брачный возраст юношах и девушках его страны. И, как положено серьезному правителю, все по науке — перед тем, как творить молодые семьи, падишах провел Глобальное тестирование и по 100-балльной шкале определил совместимость всех юношей и девушек в совместном браке.

А дальше что? Падишах наслышан про задачу о назначении, но ему не нравится ее установка. Действительно, может ли быть спокойна его душа даже в случае всеобщего благополучия, если кому-то из подданных плохо? И можно ли жертвовать интересами хотя бы одной семьи во благо общества? Конечно, нет!

Падишаху милее другая мысль. Он хочет создать максимальное число семей, причем сделать это таким образом, чтобы минимальная совместимость в семье была максимальной. А решить эту неклассическую задачу он просит вас. Помогите падишаху!

Формат входных данных

В первой строке входных данных содержатся два целых числа n и m — количество юношей и количество девушек соответственно ($1 \leqslant n, m \leqslant 200$). Последующие n строк содержат по m целых чисел от 0 до 10^9 — коэффициент совместимости соответствующей пары (меньшее значение менее способствует супружеской жизни).

Формат выходных данных

В единственную строку выходного файла выведите наименьший искомый балл, при котором возможно создание максимально возможного количества семейных пар.

Пример

стандартный ввод	стандартный вывод
3 4 77 88 31 67 96 30 2 68 35 39 76 45	76

Задача B. Наркоконтроль

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

В маленьком городке «М» начала действовать служба наркоконтроля. Первая задача службы — выяснить, сколько наркоторговцев работает в окрестности города. Агенты службы опросили всех наркозависимых в городе и составили список случаев продажи травки, произошедших за одни сутки, с указанием места и времени наблюдения. Теперь аналитики хотят понять, сколько же на самом деле есть наркоторговцев. Из данных разведки известна максимальная скорость, с которой может двигаться наркоторговец. Аналитики просят вас узнать, какое минимальное количество наркоторговцев могли участвовать во всех зафиксированных случаях продажи травки.

Формат входных данных

На первой строке входного файла содержатся целые числа n и v — количество случаев продажи травки и максимальная скорость наркоторговца ($1 \leqslant n \leqslant 100, 1 \leqslant v \leqslant 10000$). Следующие n строк содержат описания случаев продажи травки в формате «ЧЧ:ММ x y», где ЧЧ:ММ — время продажи, x и y — координаты места, в котором продавалась травка (для простоты будем считать, что всё происходило на плоскости). Координаты по модулю не превышают 1000. Скорость выражена в км/ч, координаты — в км.

Формат выходных данных

Выведите в выходной файл одно число — минимальное возможное количество наркоторговцев.

Пример

стандартный ввод	стандартный вывод
4 1 12:00 0 0 13:10 0 1 14:00 1 0 15:00 1 1	2

Задача C. День рождения

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

Митя знаком с m юношами и n девушками и хочет пригласить часть из них на свой день рождения. Ему известно, с какими девушками знаком каждый юноша, и с какими юношами знакома каждая девушка. Он хочет добиться того, чтобы каждый приглашённый был знаком со всеми приглашёнными противоположного пола, пригласив при этом максимально возможное число своих знакомых. Помогите ему это сделать!

Формат входных данных

Входной файл состоит из одного или нескольких наборов входных данных. В первой строке входного файла записано число наборов k ($1 \leqslant k \leqslant 20$). В последующих строках записаны сами наборы входных данных.

В первой строке каждого набора задаются числа $0 \leqslant m \leqslant 150$ и $0 \leqslant n \leqslant 150$. Далее следуют m строк, в каждой из которых записано одно или несколько чисел — номера девушек, с которыми знаком i-й юноша (каждый номер встречается не более одного раза). Строка завершается числом 0.

Формат выходных данных

Для каждого набора выведите четыре строки. В первой из них выведите максимальное число знакомых, которых сможет пригласить Митя. В следующей строке выведите количество юношей и количество девушек в максимальном наборе знакомых. Следующие две строки должны содержать номера приглашённых юношей и приглашённых девушек соответственно. Если максимальных наборов несколько, то выведите любой из них.

Пример

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

Задача D. 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 добираться 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 ($3 \leqslant N \leqslant 100000$, $1 \leqslant M \leqslant 100000$). 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 E. Компоненты вершинной двусвязности

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Компонентой вершинной двусвязности графа $\langle V, E\rangle$ называется максимальный по включению подграф (состоящий из вершин и ребер), такой что любые два ребра из него лежат на вершинно простом цикле.

Дан неориентированный граф без петель. Требуется выделить компоненты вершинной двусвязности в нем.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и ребер графа соответственно ($1 \leqslant n \leqslant 20\,000$, $1 \leqslant m \leqslant 200\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leqslant b_i, e_i \leqslant n$).

Формат выходных данных

В первой строке выходного файла выведите целое число k — количество компонент вершинной двусвязности графа.

Во второй строке выведите m натуральных чисел $a_1, a_2, \ldots, a_m$, не превосходящих k, где a_i — номер компоненты вершинной двусвязности, которой принадлежит i-е ребро. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Компоненты требуется нумеровать в порядке возрастания минимального номера ребра, входящего в компоненту.

Пример

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

Задача F. Возвращение домой

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

С наступлением новых ограничений из-за пандемии в Лёшином вузе снова ввели дистанционное обучение, поэтому он решил наконец-то съездить домой к родителям. К сожалению, эта поездка всегда занимает у него много времени, потому что ему сначала приходится лететь на самолетё в Москву, а потом лететь ещё столько же до своего родного города.

Всего в стране n городов и m прямых двусторонних рейсов. Между парой городов может быть несколько двусторонних рейсов, может существовать двусторонний рейс между одним и тем же городом (некоторым людям просто нравится летать и читать книжку).

Сидя в самолёте Лёша подумал, как было бы хорошо если бы рейс из города A в город B, а после из города B в город C превратился бы в один прямой рейс между городами A и C, а рейсы AB и BC просто бы исчезли (сокращение AB означает рейс между городами A и B).

Так как Лёша любит решать различные задачи, а в самолете было скучно, он решил узнать можно ли, зная все рейсы в стране и последовательно используя операцию превращения существующих рейсов AB и BC в рейс AC с удалением рейсов AB и BC, оставить всего один рейс в стране. При мысленном выполнении такой операции Лёша может выбирать любые два различных рейса, в том числе два двусторонних рейса между одной и той же парой городов или рейс между городом и им самим. Если при выполнении операции должен исчезнуть рейс AB, а таких рейсов несколько, то рейс AB удаляется столько раз, сколько он используется в этой операции (то есть один или два).

Помогите Леше по заданным исходным рейсам между городами узнать можно ли при помощи применения произвольное число раз операции, описанной выше, оставить всего один рейс.

Формат входных данных

В первой строке через пробел записаны три числа n, m и p ($1 \leqslant n, m \leqslant 10^5$, $p \in \{0, 1\}$) — количество городов в стране, количество двусторонних рейсов между городами и нужно ли выводить последовательность операций, если можно оставить ровно один рейс.

В следующих m строках находятся пары вершин a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$) — номера городов, между которыми проходит i-й рейс.

Формат выходных данных

В первой строке выведите «NO», если нельзя оставить ровно один рейс. В противном случае выведите «YES».

Если $p = 0$, то не нужно выводить последовательность операций при существовании ответа. Если же $p = 1$, то далее выведите $m - 1$ строк, в каждой строке через пробел должны быть записаны числа a_i, b_i и c_i ($1 \leqslant a_i, b_i, c_i \leqslant n$), означающие номера городов, для которых рейсы a_ib_i и b_ic_i превращаются в a_ic_i, а сами рейсы удаляются.

Примеры

стандартный ввод	стандартный вывод
3 2 1 1 2 2 3	YES 3 2 1
3 3 1 1 2 2 3 1 3	YES 1 2 3 1 3 1
3 3 0 1 2 2 3 1 3	YES
4 6 1 1 2 2 3 3 4 4 1 1 3 2 4	NO

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп.

Группа	Баллы	Дополнительные ограничения		Треб. группы	Комментарий
		n, m	p		
0	0	–	–	–	Sample tests.
1	10	$n, m \leqslant 5$	–	0	
2	20	$n, m \leqslant 1000$	–	0, 1	
3	15	$n, m \leqslant 100\,000$	–	–	Каждый город имеет не более 2 рейсов (если рейс идет из города в него самого, то он сам по себе рассматривается как 2 рейса для этого города)
4	20	$n, m \leqslant 100\,000$	$p = 0$	–	Вам нужно только вывести «NO»/«YES», последовательность операций выводить не нужно даже в случае положительного ответа
5	35	–	–	0 – 4	

Задача G. 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 контест 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 ($1 \leqslant t \leqslant 500$).

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

В каждой из последующих m строк содержится упорядоченная пара чисел a и b ($1 \leqslant a, b \leqslant n$), 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 3 2 4 1 3 3 2 2 2 1 2 2 1 3 1 2 3 4 4 1 2 2 4 3 4 1 3	4 1 3 2 4 0 0 2 1 4

Задача H. Экспедиция

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

Винни-Пух отправился в очередную Экспедицию по Лесу!

Лес представляет собой n полянок, соединённых m двусторонними тропинками. При этом возможно, что для некоторых полянок не существует способа добраться из одной в другую по тропинкам. За половину суток (день или ночь) Пух проходит ровно по одной тропинке.

Когда Пух пришёл на рассвете на очередную полянку, ему показалось, что он вернулся к полянке, с которой он начал свое путешествие. Он также уверен, что не проходил ни по одной тропинке дважды, и что все полянки до этого были различными. Также он помнит, что он отправился в Экспедицию на рассвете, то есть число дней в пути было равно числу ночей.

Вы решили помочь медвежонку и хотите определить по карте Леса, могло ли такое случиться.

Формат входных данных

Первая строка содержит число T — число наборов входных данных, которые вам предстоит обработать. Далее следуют T наборов входных данных. Каждый набор задаётся в следующем формате.

В первой строке набора входных данных заданы два числа n и m ($2 \leqslant n \leqslant 500\,000$, $1 \leqslant m \leqslant 500\,000$) — количество полянок и количество тропинок в соответствующем Лесу.

В следующих m строках содержатся описания тропинок: каждая строка состоит из двух целых чисел u_i, v_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$), обозначающих номера полянок, соединённых очередной тропинкой.

Гарантируется, что никакие две полянки не соединены двумя или более тропинками.

Гарантируется, что и суммарное число полянок, и суммарное число тропинок по всем наборам входных данных не превзойдёт 500 000.

Формат выходных данных

Для каждого набора входных данных выведите в отдельной строке «`YES`», если в Лесу существует маршрут, удовлетворяющий описанию Пуха, и «`NO`» в противном случае.

Пример

стандартный ввод	стандартный вывод
2 5 6 1 2 2 3 3 4 4 5 5 2 5 3 3 3 1 2 2 3 3 1	YES NO

Замечание

Тест из условия состоит из двух наборов входных данных.

В первом примере Пух мог начать Экспедицию на полянке с номером 2, в первый день перейти на полянку 5, в первую ночь перейти на полянку 3, во второй день перейти на полянку 4 и во вторую ночь вернуться обратно на полянку 2.

Во втором примере, независимо от выбора начальной полянки, Пух может вернуться на неё, но ему в любом случае потребуется три перехода, а значит дней в пути будет больше, чем ночей.

Задача I. Граф

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.75 секунд
Ограничение по памяти:	256 мегабайт

Дан неориентированный граф, в котором каждое ребро покрашено либо в чёрный цвет, либо в красный цвет.

Ваша задача присвоить каждой вершине по вещественному числу таким образом, что:

- сумма чисел на обеих концах каждого чёрного ребра равна 1;
- сумма чисел на обеих концах каждого красного ребра равна 2;
- сумма модулей всех присвоенных чисел наименьшая возможная.

Если это не возможно, выведите, что не существует такого комплекта чисел

Формат входных данных

Первая строка содержит два целых числа N ($1 \leqslant N \leqslant 100\,000$) и M ($0 \leqslant M \leqslant 200\,000$) — количество вершин и рёбер, соответственно. Вершины пронумерованы последовательными числами от 1 до N.

Следующие M строк содержат описание рёбер. Каждая строка содержит три целых числа a, b и c, которые обозначают, что между вершинами a и b ($1 \leqslant a, b \leqslant N$) есть ребро цвета c (1 обозначает чёрное, 2 обозначает красное).

Формат выходных данных

Если решение существует, выведите в первой строке слово «`YES`» и во второй строке выведите N чисел. Для каждого i ($1 \leqslant i \leqslant N$), i-тое из этих чисел должно равняться числу, присвоенному вершине с номером i.

Вывод должен соответствовать следующим ограничениям точности:

- сумма чисел на концах каждого ребра должна отличаться от нужной суммы для этого ребра меньше, чем на 10^{-6};
- сумма модулей всех присвоенных чисел отличается от наименьшего возможного меньше, чем на 10^{-6}.

Если существует несколько решений, выведите любое из них. Если решения не существует, выведите одно слово «`NO`».

Примеры

стандартный ввод	стандартный вывод
4 4 1 2 1 2 3 2 1 3 2 3 4 1	YES 0.5 0.5 1.5 -0.5
2 1 1 2 1	YES 1.0 0.0
3 2 1 2 2 2 3 2	YES 0.0 2.0 0.0
3 4 1 2 2 2 2 1 2 1 1 1 2 2	NO

Задача J. Трафик

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

Центр Гдыни находится на острове по середине реки Кача. Каждое утро тысячи машин проезжают через этот остров из спальных районов на западном берегу реки к индустриальным районам на восточном берегу.

Остров представляет из себя прямоугольник, стороны которого параллельны осям координат. Представим его, как прямоугольник $A \times B$, противоположные углы которого находятся в точках $(0, 0)$ и (A, B). На острове есть n перекрестков, пронумерованных от 1 до n. Перекресток с номером i находится в точке (x_i, y_i). Если перекресток находится в точках вида $(0, y)$, он находится на западной части острова, а если в точках вида (A, y) — на восточной. Перекрестки соединены улицами. Каждая улица — это отрезок, соединяющий два перекрестка. Улицы бывают как односторонние, так и двусторонние. Никакие улицы не пересекаются, кроме как в перекрестках, которые являются концами улиц. На острове нет мостов и туннелей. Из-за растущей загруженности дорог мэр города нанял вас проверить, сколько перекрестков в восточной части острова достижимы по улицам из каждого перекрестка западной части.

Формат входных данных

Первая строка содержит четыре целых числа n, m, A и B ($1 \leqslant n \leqslant 3 \cdot 10^5$, $0 \leqslant m \leqslant 9 \cdot 10^5$, $1 \leqslant A, B \leqslant 10^9$) — число перекрестков, число улиц и размеры острова, соответственно.

В каждой из следующих n строк содержится по два целых числа x_i, y_i ($0 \leqslant x_i \leqslant A$, $0 \leqslant y_i \leqslant B$), которые описывают координаты перекрестка i. Никакие два перекрестка не находятся в одной точке.

Следующие m строк описывают улицы. Каждая улица описывается тремя целыми числами c_i, d_i, k_i ($1 \leqslant c_i, d_i \leqslant n$, $c_i \neq d_i$, $k_i \in \{1, 2\}$), описывающими улицу, соединяющую перекрестки c_i и d_i. Причем, если $k_i = 1$, то улица односторонняя из c_i в d_i, а иначе по улице можно ездить в обоих направлениях. Каждая упорядоченная пара (c_i, d_i) встречается не более одного раза.

Гарантируется, что существует хотя бы один перекресток в западной части, из которого можно добраться до какого-нибудь перекрестка восточной части острова.

Формат выходных данных

Выведите по одной строке для каждого перекрестка на западной стороне острова, в порядке убывания y-координаты перекрестка.

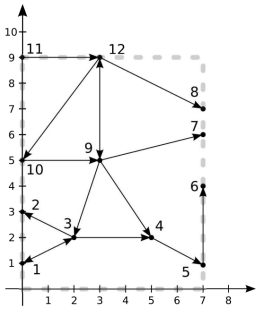
Строка должна содержать число перекрестков левой стороны острова, достижимых из соответствующего перекрестка западной стороны.

Примеры

стандартный ввод	стандартный вывод
5 3 1 3 0 0 0 1 0 2 1 0 1 1 1 4 1 1 5 2 3 5 2	2 0 2
12 13 7 9 0 1 0 3 2 2 5 2 7 1 7 4 7 6 7 7 3 5 0 5 0 9 3 9 1 3 2 3 2 1 3 4 1 4 5 1 5 6 1 9 3 1 9 4 1 9 7 1 9 12 2 10 9 1 11 12 1 12 8 1 12 10 1	4 4 0 2

Замечание

Картинка ко второму примеру



Задача K. История о кротах

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	1024 мегабайта

Крот — маленький тщедушный зверек...

Примечание переводчика

Не так давно семья кротов решила прокопать новую сеть туннелей под землей. План сети, который уже разработан, состоит из подземных комнат и туннелей, которые их соединяют. Формально, план сети представляет собой неориентированный связный граф. Кротиха-мать хочет использовать данную работу как возможность научить ее двоих детей-кротов, как правильно вырыть сеть туннелей.

В качестве начальной быстрой демонстрации кротиха-мать собирается вырыть несколько комнат и туннелей, которые образуют простой путь в графе плана сети. После этого она разделит оставшиеся комнаты между своими двумя детьми так, чтобы каждому из детей досталось равное количество комнат, иначе они могут расстроиться (насчет туннелей можно не беспокоиться, потому что их копать гораздо проще). Дети могут работать в любых из своих комнат в любом порядке, в котором они захотят.

Так как у детей нет опыта по копанию сети туннелей, кротиха-мать поняла одну проблему: если между некоторой парой комнат, которые принадлежат разным детям, в плане есть туннель, то существует риск происшествия во время выкапывания данного туннеля.

Помогите кротихе-матери решить, какой путь использовать для начальной демонстрации, и как разделить оставшиеся комнаты поровну так, чтобы не существовало пары комнат, отданных разным детям, соединенных туннелем. Изначальный путь должен содержать хотя бы одну комнату и не должен посещать какую-либо комнату два раза.

Формат входных данных

В первой строке записаны два целых числа c и t ($1 \leqslant c \leqslant 2 \cdot 10^5$, $0 \leqslant t \leqslant 2 \cdot 10^5$) — количество комнат и туннелей в плане.

В каждой из следующих t строк записаны два целых числа a и b ($1 \leqslant a, b \leqslant c$, $a \neq b$) — номера комнат, соединенных туннелем.

Комнаты пронумерованы числами от 1 до c. Между каждой парой комнат есть не более одного туннеля. Между любой парой комнат существует путь.

Формат выходных данных

В первой строке выведите два целых числа p и s — количество комнат в пути, выбранном кротихой-матерью для демонстрации, и количество комнат, которые должен быть вырыть каждый из детей.

Во второй строке выведите p чисел — номера комнат, образующих путь для демонстрации. Комнаты должны быть выведены в порядке следования по пути.

В третьей строке выведите s чисел — номера комнат, отданных первому ребенку в произвольном порядке.

В четвертой строке выведите s чисел — номера комнат, отданных второму ребенку в произвольном порядке.

Гарантируется, что ответ существует. Если существует несколько корректных ответов, выведите любой.

Примеры

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

Задача L. Оплачивайте проезд

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

В стране JOI n городов, пронумерованных от 1 до n. Город с номером 1 — столица страны. В стране есть всего одна железнодорожная компания, которая обслуживает m двусторонних железных дорог, пронумерованных от 1 до m. Железная дорога с номером i соединяет города u_i и v_i.

Вы не можете перемещаться между городами, кроме как по железной дороге. Гарантируется, что от любого города возможно добраться до любого другого.

Сейчас плата за проезд по любой железной дороге — одна йена. В течении следующих q лет планируется увеличивать стоимость проезда. Согласно этому плану в j-й год стоимость проезда по железной дороге r_j увеличится с одной до двух йен. Каждая железная дорога встречается в плане не более одного раза.

Между тем, каждый год эта компания проводит опрос населения в каждом городе, с помощью которого выясняет, насколько жители города удовлетворены услугами, предоставляемыми компанией. Изначально все жители городов удовлетворены компанией, но, при повышении стоимости проезда, их мнение может поменяться.

Каждый год опрос производится после увеличения стоимости проезда по железной дороге, соответствующей этому году. То есть, во время проведения опроса в год j, стоимости проездов по всем железным дорогам r_i, где $i \leqslant j$, увеличены, а по остальным — не изменены. Жители города k ($2 \leqslant k \leqslant n$) в год j будут удовлетворены в том и только том случае, если:

- Минимальная стоимость проезда от города k до столицы не изменилась с начала проведения плана.

Стоимость последовательного проезда по нескольким железным дорогам равна сумме стоимостей проезда по ним. Жители столицы никогда не станут неудовлетворенны услугами компании. Обратите внимание, что путь, на котором достигается минимальная стоимость проезда от города k до столицы, может поменяться с начала проведения плана.

Вычислите количество городов, жители которых будут неудовлетворенны услугами компании, после каждого из q изменений.

Формат входных данных

В первой строке входного файла содержится три целых числа n, m и q ($2 \leqslant n \leqslant 100\,000$, $1 \leqslant q \leqslant m \leqslant 200\,000$) — количество городов, железных дорог и лет, в течении которых планируется проводить план.

В следующих m строках содержится по два целых числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$) — города, которые соединяет i-я железная дорога.

Гарантируется, что в графе нет петель и кратных ребер. От каждого города возможно добраться до столицы по железным дорогам.

В следующих q строках содержится по одному целому числу r_j ($1 \leqslant r_j \leqslant m$) — номер железной дороги, стоимость проезда по которой планируется увеличить на j-й год. Все r_j различны.

Формат выходных данных

Выведите q чисел, по одному на строке, — количество городов, жители которых недовольны железнодорожной компанией после j-го изменения.

Система оценки

Подзадача 1 [12 баллов]:

- $n \leqslant 100$
- $m \leqslant 4\,950$

- $q \leqslant 30$

Подзадача 2 [14 баллов]:

- $q \leqslant 30$

Подзадача 3 [35 баллов]:

- В ответе будет не более 50 различных чисел

Подзадача 4 [39 баллов]:

- Нет дополнительных ограничений

Примеры

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

Задача M. Путешествие по музеям

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

В стране N есть n городов, соединённых m односторонними дорогами. Эта, казалось бы, неприметельная страна всё же интересна двумя своими особенностями. Во-первых, неделя здесь длится d дней. Во-вторых, в каждом городе страны N расположен ровно один музей.

Турфирма «Открытые музеи» разрабатывает новую программу для путешественников, интересующихся музеями. Работникам турфирмы известно, в какие дни недели каждый из музеев открыт для посещения. Путешествие должно начаться в столице — городе с номером 1, причём день начала путешествия должен быть первым днём недели. Днём турист будет находиться в городе и смотреть экспозицию музея (в случае если музей, конечно, работает сегодня), а в конце дня путешествие либо заканчивается, либо турист отправляется в другой город, соединённый с текущим дорогой. Дорожная система N устроена так, что перемещение по одной дороге всегда занимает одну ночь, кроме того все дороги в стране **односторонние**. Во время путешествия разрешается посещать один город несколько раз.

Вам требуется разработать такой маршрут для путешествия, что количество **различных** музеев, которые можно посетить за его время, было бы как можно больше.

Формат входных данных

Первая строка содержит три целых числа n, m и d ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 100\,000$, $1 \leqslant d \leqslant 50$) — количество городов, количество дорог и количество дней в неделе.

Каждая из следующих m строк содержит два целых числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$), обозначающие **одностороннюю** дорогу из города u_i в город v_i.

Следующие n строк содержат график работы музеев. График работы музея, находящегося в i-м городе, описывается в i-й из этих строк. Каждая строка состоит ровно из d символов «`0`» или «`1`», j-й символ строки равен «`1`», если музей открыт в j-й день недели, и «`0`» в противном случае.

Гарантируется, для каждой пары городов (u, v) существует не более одной дороги, ведущей из u в v.

Формат выходных данных

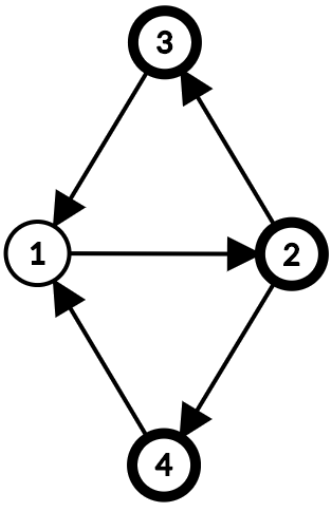
Выведите одно целое число — максимальное количество различных музеев, которые можно посетить, начав путешествие в первом городе в первый день недели.

Примеры

стандартный ввод	стандартный вывод
4 5 3 3 1 1 2 2 4 4 1 2 3 011 110 111 001	3
3 3 7 1 2 1 3 2 3 1111111 0000000 0111111	2

Замечание

Пояснение к первому примеру



Максимально возможное количество музеев для посещения равно 3. Можно посетить 3 различных музея, например, так, как описано ниже.

День 1. Сейчас день недели 1, и турист находится в городе 1. Музей там закрыт. Ночью турист направляется в город 2.

День 2. Сейчас день недели 2, и турист находится в городе 2. Музей там открыт, турист его посещает. Ночью турист направляется в город 4.

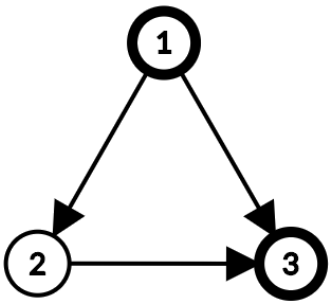
День 3. Сейчас день недели 3, и турист находится в городе 4. Музей там открыт, турист его посещает. Ночью турист направляется в город 1.

День 4. Сейчас день недели 1, и турист находится в городе 1. Музей там закрыт. Ночью турист направляется в город 2.

День 5. Сейчас день недели 2, и турист находится в городе 2. Музей там открыт, но турист в нём уже был. Ночью турист направляется в город 3.

День 6. Сейчас день недели 3, и турист находится в городе 3. Музей там открыт, турист его посещает. На этом путешествие заканчивается.

Пояснение ко второму примеру



Максимально возможное количество музеев для посещения равно 2. Можно посетить 2 различных музея, например, так, как описано ниже.

День 1. Сейчас день недели 1, и турист находится в городе 1. Музей там открыт, турист его посещает. Ночью направляемся в город 2.

День 2. Сейчас день недели 2, и турист находится в городе 2. Музей там закрыт. Ночью направляемся в город 3.

День 3. Сейчас день недели 3, и турист находится в городе 3. Музей там открыт, турист его посещает. На этом путешествие заканчивается.

Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		n	d	График работы		
0	0	–	–	–	–	Тесты из условия.
1	9	$n \leqslant 10$	$d \leqslant 10$	–	0	
2	12	$n \leqslant 100$	$d \leqslant 10$	–	0, 1	
3	13	–	–	Каждый музей **работает** ровно один день в неделе	–	
4	17	–	–	Каждый музей **не работает** ровно один день в неделе	–	
5	13	–	$d = 2$	–	–	
6	36	–	–	–	0, 1, 2, 3, 4, 5	**Offline-проверка**

Задача N. Xor the Graph

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

Дан неориентированный граф, состоящий из N вершин и M рёбер. Каждому ребру присвоено значение, 0 или 1. Операция состоит в том, чтобы выбрать путь (не обязательно простой), и изменить значения, записанные на каждом его ребре на противоположные. Сделайте значения на всех рёбрах нулями за минимальное число операций.

Формат входных данных

На первой строке заданы два целых числа N и M ($1 \leqslant N \leqslant 10^5$, $0 \leqslant M \leqslant 10^5$).

В каждой из следующих M строк записаны по три целых числа a, b и c. Они означают, что существует ребро между вершинами (a, b), изначальное значение которого равно c ($1 \leqslant a, b \leqslant n$, $0 \leqslant c \leqslant 1$).

Формат выходных данных

Выведите количество операций K на первой строке.

В последующих K строках выведите пути, которые вы выбрали:

- Первым числом выведите P — количество вершин в пути
- Затем выведите P целых чисел $v_1, v_2, \ldots, v_P$ — индексы вершин. Для всех $1 \leqslant i \leqslant P - 1$ (v_i, v_{i+1}) должно являться ребром в заданном графе.

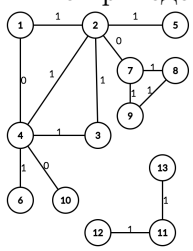
Решение будет считаться верным, если оно использует минимальное количество операций, и при этом $\sum P \leqslant 4 * M$.

Пример

стандартный ввод	стандартный вывод
13 14 1 2 1 2 3 1 3 4 1 2 4 1 1 4 0 4 6 1 4 10 0 2 5 1 2 7 0 7 8 1 8 9 1 9 7 1 11 12 1 11 13 1	3 5 1 2 3 4 6 8 4 2 7 8 9 7 2 5 3 12 11 13

Замечание

Ниже приведён граф в первом тесте.



Задача O. Сбалансированная строка

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

Надо найти лексикографически минимальную строку s длины n, состоящую из 0 и 1, что она удовлетворяет m ограничениям: на отрезке $[L_i, R_i]$ должно быть равное количество нулей и единиц. Гарантируется, что длина каждого отрезка чётная.

Формат входных данных

Первая строка содержит два числа n и m ($2 \leqslant n \leqslant 10^6, 1 \leqslant m \leqslant 2 \cdot 10^5$) — длина строки и количество ограничений соответственно.

Следующие m строк содержат отрезки L_i и R_i ($1 \leqslant L_i < R_i \leqslant n, R_i - L_i + 1 \equiv 0 \bmod 2$).

Формат выходных данных

Выведите лексикографически минимальную строку, удовлетворяющую этим требованиям.

Примеры

стандартный ввод	стандартный вывод
4 2 1 2 3 4	0101
6 2 1 4 3 6	001100
20 10 6 17 2 3 14 19 5 14 10 15 7 20 10 19 3 20 6 9 7 12	00100100101101001011

Задача P. Двойная сортировка II

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

Даны две перестановки a и b, обе размера n. Перестановка размера n — это массив из n элементов, в который каждое целое число от 1 до n входит ровно один раз. Элементы в каждой перестановке пронумерованы от 1 до n.

Вы можете выполнять следующую операцию любое количество раз:

- выбрать целое число i от 1 до n;
- обозначим за x такое целое число, что $a_x = i$. Поменять местами a_i и a_x;
- обозначим за y такое целое число, что $b_y = i$. Поменять местами b_i и b_y.

Вы должны отсортировать обе перестановки **в порядке возрастания** (то есть должны выполняться условия $a_1 < a_2 < \cdots < a_n$ и $b_1 < b_2 < \cdots < b_n$) за **минимальное количество операций**. Обратите внимание, что обе перестановки должны быть отсортированы после того, как вы выполните выбранную вами последовательность операций.

Формат входных данных

В первой строке задано одно целое число n ($2 \leqslant n \leqslant 3000$).

Во второй строке заданы n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant n$; все a_i различны).

В третьей строке заданы n целых чисел $b_1, b_2, \ldots, b_n$ ($1 \leqslant b_i \leqslant n$; все b_i различны).

Формат выходных данных

Сначала выведите одно целое число k ($0 \leqslant k \leqslant 2n$) — минимальное количество операций, за которое можно отсортировать обе перестановки. Обратите внимание: можно показать, что $2n$ операций всегда достаточно.

Затем выведите k целых чисел $op_1, op_2, \ldots, op_k$ ($1 \leqslant op_j \leqslant n$), где op_j — значение i, выбранное для j-й операции.

Если ответов несколько, выведите любой из них.

Примеры

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

Задача Q. Pink Floyd

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

Это интерактивная задача.

Учёные находятся всего в нескольких шагах от открытия новой оптимизации алгоритма Флойда, позволяющей ему работать за линейное время! Осталась лишь одна часть.

Как известно, для запуска алгоритма Флойда в графе на n вершинах должно быть по одному ребру между любыми двумя вершинами. У ученых есть такой граф, однако, они ориентировали каждое ребро, то есть направили его ровно в одну из двух сторон.

Для оптимизации Флойда выбраны m рёбер и покрашены в розовый цвет, остальные ребра покрашены в зеленый. Вы знаете направление каждого из этих m ребер, однако, направление зеленых рёбер вам заранее неизвестно. За один запрос вы можете узнать направление одного зеленого ребра, но всего вы можете сделать не более $2 \cdot n$ запросов.

Вам нужно найти такую вершину, от которой любая другая вершина достижима по пути, состоящем из ребер одного цвета. Обратите внимание, ученые могут лукавить, что уже определили направления для всех ребер, и выдавать ответы в зависимости от ваших запросов.

Формат входных данных

В первой строке находятся два целых числа n и m ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 100\,000$) — число вершин и число рёбер розового цвета.

В i-й из следующих m строках находятся два целых числа u_i, v_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$) — начало i-го ребра розового цвета и его конец. Гарантируется, что все неупорядоченные пары (u_i, v_i) различны.

Формат выходных данных

Когда вы найдете ответ, выведите «`!`» и номер вершины, от которой каждая другая достижима по одноцветному пути.

Протокол взаимодействия

Чтобы узнать направление зеленого ребра между вершинами a и b, в одной строке через пробел выведите символ «`?`», а затем два целых числа a и b.

В ответ на это считайте целое число, которое будет равно 1, если ребро направленно от a к b, и 0, если ребро направлено от b к a.

Вы можете задать не более $2 \cdot n$ вопросов, в противном случае вы получите вердикт `Wrong Answer`.

После вывода запроса не забудьте вывести перевод строки и сбросить буфер вывода. В противном случае вы получите вердикт `Решение «зависло»`. Для сброса буфера используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- смотрите документацию для других языков.

Ответ -1 вместо 0 или 1 означает, что ваша программа сделала некорректный запрос или превысила число запросов. Ваша программа должна немедленно завершиться после прочтения ответа -1, вы получите вердикт `Неправильный ответ`. В противном случае вы можете получить любой вердикт, так как программа продолжит чтение из закрытого потока.

Взломы

Для взлома задайте тест в следующем формате. В первой строке должны находиться два целых числа n и m ($1 \leqslant n \leqslant 300$, $0 \leqslant m \leqslant n \cdot (n-1)/2$) — число вершин и число рёбер розового цвета.

В i-й из следующих m строк должны находиться два целых чисел a_i, b_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$)), означающих, что есть ребро розового цвета от вершины a_i к вершине b_i. Все неориентированные пары (a_i, b_i) должны быть различны.

В i-й из следующих $(n \cdot (n-1)/2 - m)$ строках должны находиться два целых числа a_i, b_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$)), означающих, что есть ребро зеленого цвета от вершины a_i к вершине b_i. Все неориентированные пары (a_i, b_i) должны быть различны и не должны совпадать с теми же парами для рёбер розового цвета.

Пример

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

Замечание

В приведённом примере ответ на запрос «? 1 3» — 0, поэтому ребро ориентированно от 3 к 1; ответ на запрос «? 4 2» — 1, поэтому ребро ориентированно от 4 к 2; ответ на запрос «? 3 2» — 1, поэтому ребро ориентированно от 3 к 2. Тогда от вершины 3 есть пути по зелёным рёбрам к вершинам 1 и 2, а так же есть путь по розовым рёбрам до вершины 4.

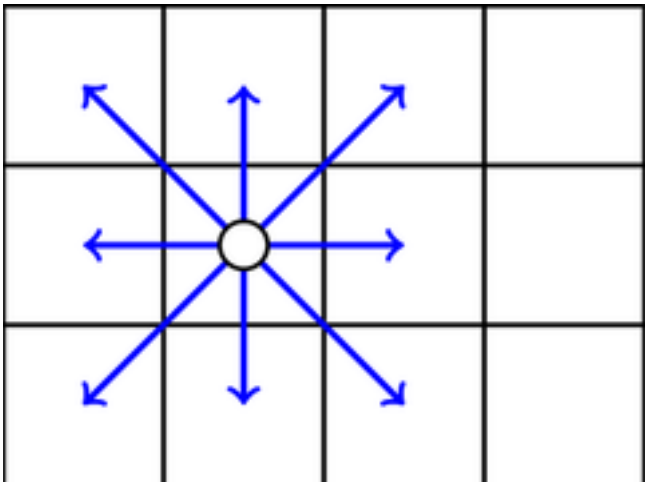
Задача R. Полезные ископаемые

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

Ведется проект по освоению планеты соседней звездной системы. Для добычи полезных ископаемых планируется направить на планету несколько партий роботов.

Участок поверхности планеты, на котором планируется добывать полезные ископаемые, представляет собой клетчатый прямоугольник размером w на h, клетки участка имеют координаты от $(1, 1)$ до (w, h). В некоторых клетках участка находятся базы специалистов, в которые могут быть доставлены партии роботов. Всего на участке размещено s баз, и i-я база находится в клетке с координатами (x_i, y_i).

Каждая партия роботов характеризуется тремя параметрами: j-я партия доставляется на базу b_j, содержит n_j роботов и каждый робот партии обладает мобильностью m_j. Когда партия роботов доставляется на соответствующую базу, каждый робот этой партии перемещается по поверхности планеты от базы до некоторой клетки. Если мобильность робота равна m, он может не более m раз переместиться на одну из восьми соседних клеток, как показано на рис. 1.



После того как роботы из всех доставленных партий размещаются на участке, они активируются и начинают добычу полезных ископаемых. В процессе перемещения в одной клетке может одновременно находиться произвольное количество роботов. Однако после активации в каждой клетке должно находиться не более q роботов.

Руководством проекта получена информация о t партиях роботов, которые могут быть последовательно отправлены на планету. После доставки всех партий роботов, учитывая их ограниченную мобильность, возможна ситуация, что не удастся разместить роботов на участке так, чтобы в каждой клетке оказалось не больше q роботов. Поэтому руководство должно выбрать k первых партий роботов, где $0 \leqslant k \leqslant t$, которые будут полностью доставлены на соответствующие базы. После этого, если $k < t$, следует дополнительно принять z из n_{k+1} роботов следующей, $(k + 1)$-й партии, $0 \leqslant z < n_{k+1}$.

Все полученные таким образом роботы должны с учетом ограничения на мобильность разместиться на участке таким образом, чтобы в каждой клетке было не более q роботов. После этого они будут активированы и начнут добычу полезных ископаемых. Разумеется, руководство проекта старается максимизировать количество роботов, которые будут доставлены на планету, поэтому, с учетом описанных ограничений, требуется максимизировать k, а затем максимизировать z.

Требуется написать программу, которая по размерам участка, числу q, описанию расположения баз, а также количеству запланированных партий роботов и их описанию определяет максимальное число k — количество партий роботов, и затем – максимальное число z – дополнительное количество роботов из $(k + 1)$-й партии, чтобы, доставив роботов на планету, их можно было разместить на участке таким образом, чтобы в каждой клетке оказалось не более q роботов.

Формат входных данных

Первая строка входного файла содержит числа w, h, s и q ($1 \leqslant w, h \leqslant 10^5$, $1 \leqslant s \leqslant 4$, $1 \leqslant q \leqslant 100$). Последующие s строк содержат по два целых числа x_i, y_i и описывают базы специалистов ($1 \leqslant x_i \leqslant w$, $1 \leqslant y_i \leqslant h$).

Следующая строка содержит число t – количество партий роботов ($1 \leqslant t \leqslant 100$). Последующие t строк описывают партии роботов и содержат по 3 целых числа: b_j, n_j и m_j ($1 \leqslant b_j \leqslant s, 1 \leqslant n_j \leqslant w \cdot h \cdot q$, $0 \leqslant m_j < \max(w, h)$).

Формат выходных данных

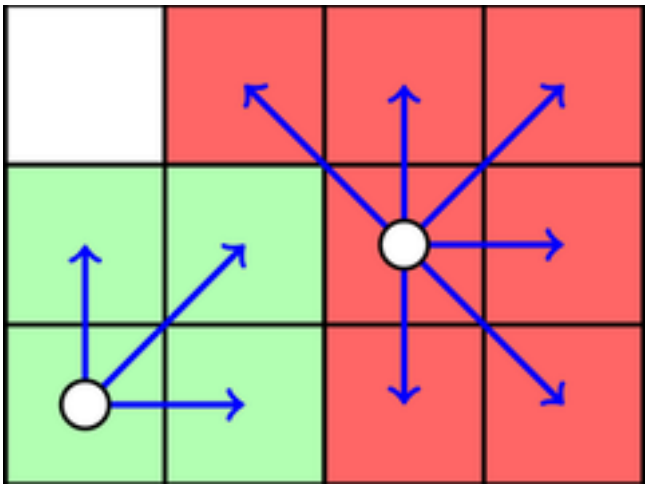
Требуется вывести два числа: k и z, $0 \leqslant k \leqslant t$. Если $k = t$, то z должно быть равно 0, иначе должно выполняться условие $0 \leqslant z < n_{k+1}$.

Пример

стандартный ввод	стандартный вывод
4 3 2 1 1 1 3 2 3 1 4 1 2 9 1 1 12 2	1 7

Замечание

В приведенном примере описания входных данных следует полностью принять первую партию роботов и дополнительно принять 7 роботов из второй партии. На рис. 2 показано, как можно разместить этих роботов на участке, чтобы в каждой клетке было не более одного робота. Базы специалистов показаны кружками. Клетки, в которых окажутся роботы с базы 1, показаны зелёным, а клетки, в которых окажутся роботы с базы 2, показаны красным цветом.



Возможное размещение роботов на участке в данном примере.

Задача S. 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 ($1 \leqslant n, m, k \leqslant 100\,000$).

В каждой из последующих k строк содержатся по два целых числа a и b, обозначающих пожелания ($1 \leqslant a \leqslant n, 1 \leqslant |b| \leqslant m$). 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 T. Не вывихни шею

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

Дан граф из n вершин (n чётное) и $\frac{3}{2}n$ рёбер. Известно, что степень каждой вершины равна 3. Пусть каждая вершина покрашена в один из двух цветов. Тогда рассмотрим такую операцию:

- Для всех вершин одновременно выбирается цвет, в который покрашено большинство их соседей, после чего вершина перекрашивается в этот цвет.

Определите, существует ли такая раскраска графа в 2 цвета, которая не может быть получена после применения данной операции ровно один раз к какой-либо раскраске.

Формат входных данных

Первая строка содержит единственное целое число t ($1 \leqslant t$) — количество тестовых случаев.

Первая строка каждого теста содержит единственное целое число n ($1 \leqslant n \leqslant 5 \cdot 10^4$) — количество вершин в графе.

Далее следует $\frac{3}{2}n$ строк, каждая из которых содержит по два числа v и u ($v < u$) — концы очередного ребра. Гарантируется, что все рёбра различны.

Сумма n по всем тестам не превосходит $5 \cdot 10^4$.

Формат выходных данных

Для каждого теста выведите -1, если такой раскраски нет, или выведите саму раскраску (считаем, что цвета обозначаются буквами «w» и «b»).

Пример

стандартный ввод	стандартный вывод
2 4 1 2 1 3 1 4 2 3 2 4 3 4 10 1 2 1 3 1 4 2 3 2 4 3 5 4 5 5 6 6 7 6 8 7 9 7 10 8 9 8 10 9 10	WWBW WWBWWWBWWW

Задача U. Два проспекта

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

Для того чтобы сделать столицу Берляндии более привлекательным туристическим местом, великий король придумал следующий план: выбрать две улицы города и назвать их проспектами. Естественно, эти проспекты будут объявлены чрезвычайно важными историческими местами, что должно привлечь туристов со всего мира.

Столицу Берляндии можно представить в виде графа, вершинами которого являются перекрестки, а ребрами являются улицы, соединяющие два перекрестка напрямую. Всего в графе n вершин и m ребер, по любой улице можно двигаться в обоих направлениях, от любого перекрестка можно добраться до любого другого, перемещаясь только по улицам, каждая улица соединяет два различных перекрестка, и никакие две улицы не соединяют одинаковую пару перекрестков.

Чтобы снизить поток обычных горожан, перемещающихся по великим проспектам, было решено ввести платный проезд по каждому их них в обе стороны. Теперь за один проезд по проспекту нужно заплатить 1 тугрик. За проезд по остальным улицам платить не нужно.

Аналитики собрали выборку из k горожан, i-му из них надо ездить на работу от перекрестка a_i к перекрестку b_i. После выбора двух проспектов каждый горожанин будет добираться на работу вдоль пути, стоимость которого будет минимальна.

Для того чтобы заработать как можно больше денег, было решено выбрать в качестве двух проспектов две такие улицы, что суммарное количество тугриков, которые заплатят эти k горожан, будет максимально возможным. Помогите королю: по заданной схеме города и выборке горожан найдите, какие две улицы нужно сделать проспектами, и сколько тугриков заплатят горожане при таком выборе.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится два целых числа, первое число t ($1 \leqslant t \leqslant 10^5$) — количество наборов входных данных. Второе число g ($0 \leqslant g \leqslant 7$) — номер группы, к которой принадлежит тест. Далее следуют описания наборов входных данных.

В первой строке описания каждого набора входных данных находятся два целых числа n и m ($3 \leqslant n \leqslant 500\,000$, $n - 1 \leqslant m \leqslant 500\,000$, $m \leqslant \frac{n(n-1)}{2}$) — количество перекрестков и улиц города.

В следующих m строках содержатся описания улиц, в i-й строке находятся два целых числа s_i и f_i ($1 \leqslant s_i, f_i \leqslant n$, $s_i \neq f_i$) — номера перекрестков, которые соединяет i-я улица. Гарантируется, что никакие две улицы не соединяют одну и ту же пару перекрестков, и что от любого перекрестка можно добраться до любого другого, перемещаясь только по улицам.

В следующей строке находится единственное целое число k ($1 \leqslant k \leqslant 500\,000$) — количество горожан в выборке.

В следующих k строках содержатся описания горожан, в i-й строке находятся два целых числа a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$) — i-й горожанин едет на работу от перекрестка a_i до перекрестка b_i.

Пусть M обозначает сумму значений m по всем наборам входных данных, а K означает сумму значений k по всем наборам входных данных. Гарантируется, что $M, K \leqslant 500\,000$.

Формат выходных данных

Для каждого набора входных данных в выведите ответ на задачу.

В первой строке ответа данных выведите суммарное количество тугриков, которое заплатят горожане.

Во второй строке ответа выведите два целых числа x_1 и y_1 — номера перекрёстков, дорогу между которыми нужно сделать первым проспектом.

В третей строке ответа выведите два целых числа x_2 и y_2 — номера перекрёстков, дорогу между которыми нужно сделать вторым проспектом.

Номера перекрестков, соединенных улицей, можно выводить в произвольном порядке, каждая из двух выведенных улиц должна встречаться среди m улиц города, выбранные улицы должны быть различными.

Пример

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

Система оценки

Тесты к этой задаче состоят из 7 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов необходимых групп. Обратите внимание, что 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	Дополнительные ограничения			Необх. группы	Комментарий
		n	m	k		
0	0	–	–	–	–	Тесты из условия
1	14	$n \leqslant 20$	$m \leqslant 20$	$K \leqslant 1000$	0	$t \leqslant 100$
2	11	$n \leqslant 100$	$M \leqslant 2000$	$K \leqslant 2000$	0 – 1	
3	15	$n \leqslant 2000$	$M \leqslant 20\,000$	$K \leqslant 20\,000$	0 – 2	
4	16	–	$M \leqslant 100\,000$	$K \leqslant 100\,000$	0 – 3	
5	11	–	–	–	–	$n = m + 1$
6	19	–	–	–	–	Из каждого перекрестка выходит ровно 2 улицы
7	14	–	–	–	0 – 6	**Offline-проверка**