

Задача А. Студентам — бесплатно!

Имя входного файла:	students-free.in
Имя выходного файла:	students-free.out
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	64 мегабайта

Зал Большого галактического театра состоит из S рядов, по S мест в каждом ряду. Продажа билетов на каждый спектакль происходит по следующему принципу: первые $S^2 - N$ ценителей прекрасного приобретают билеты на любые места по их вкусу, а оставшиеся N кресел администрация бесплатно выделяет студентам, отдавая дань сложившимся традициям.

Во избежание обвинений в дискриминации по половому признаку, рассаживать студентов по этим N местам необходимо таким образом, чтобы:

- в каждом ряду количество девушек-студенток и количество юношей-студентов различалось бы не более чем на 1;
- на каждой "вертикали мест" (т. е. местах, имеющих один и тот же номер, но расположенных в разных рядах) количество девушек-студенток и количество юношей-студентов также различалось бы не более чем на 1.

Таким образом, после продажи билетов ценителям прекрасного билетёры должны распределить оставшиеся N кресел на женские и мужские с соблюдением этих правил. Каждое место в зале определяется двумя числами от 1 до S — номером ряда и номером самого места в этом ряду. Студенческое кресло номер i расположено в a_i -м ряду и имеет в нём номер b_i . Поскольку ценители прекрасного могли занять совершенно любые места, числа a_i и b_i могут принимать любые значения от 1 до S . В частности, может оказаться так, что в каком-нибудь ряду не будет ни одного студенческого места.

Ради упрощения работы билетёров администрация обращается к вам с заданием написать программу, которая автоматизирует процесс распределения студенческих мест на мужские и женские.

Формат входных данных

Сначала вводятся два целых числа S и N ($1 \leq S \leq 100\,000, 1 \leq N \leq \min\{100\,000, S^2\}$). Далее расположены N пар натуральных чисел (a_i, b_i) , не превосходящих S . Гарантируется, что все места различные.

Формат выходных данных

Если искомого способа не существует, выведите **Impossible**. Иначе выведите единственную строку из N символов М (мужское) и W (женское). Символ на i -й позиции соответствует статусу i -го места в той же нумерации, в которой они были перечислены во входных данных.

Примеры

students-free.in	students-free.out
2 2 2 1 1 2	WM
3 5 1 2 2 3 1 3 2 1 1 1	MMWWM

Задача В. Раскраска в три цвета

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

Петя нарисовал на бумаге n кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

Формат входных данных

Первая строка содержит два целых числа n и m — количество кружков и количество линий, которые нарисовал Петя, соответственно ($1 \leq n \leq 1000$, $0 \leq m \leq 20\,000$).

Следующая строка содержит n символов из множества $\{‘R’, ‘G’, ‘B’\}$ — i -й из этих символов означает цвет, в который раскрашен i -й кружок (‘R’ — красный, ‘G’ — зеленый, ‘B’ — синий).

Следующие m строк содержат по два целых числа — пары кружков, соединенных отрезками.

Формат выходных данных

Выведите в выходной файл одну строку, состоящую из n символов из множества $\{‘R’, ‘G’, ‘B’\}$ — цвета кружков после перекраски. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл слово “Impossible”.

Примеры

<code>color.in</code>	<code>color.out</code>
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	GGBR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

Задача С. Декартово

Имя входного файла:	cartesius.in
Имя выходного файла:	cartesius.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Государство Иксово состоит из N_x городов, некоторые пары которых связаны дорогами с двусторонним движением. Каждая дорога имеет свою длину. Всего межгородских дорог в стране M_x , при чем известно, что из каждого города Иксевщины можно доехать по дорогам до каждого другого города этой страны. Города Иксово пронумерованы натуральными числами от 1 до N_x .

Государство Игреково состоит из N_y городов, некоторые пары которых связаны дорогами с двусторонним движением. Каждая дорога имеет свою длину. Всего межгородских дорог в стране M_y , при чем известно, что из каждого города Игреково можно доехать по дорогам до каждого другого города этой страны. Города Игреково пронумерованы натуральными числами от 1 до N_y .

Страна Декартово состоит из $N = N_x \cdot N_y$ городов: каждому городу Декартово во взаимно однозначное соответствие можно поставить пару городов-побратимов (x, y) , где x — город Иксово, а y — город Игреково. Некоторые пары городов Декартово также соединены дорогами с двусторонним движением. Дорог в стране ровно $M = N_x \cdot M_y + N_y \cdot M_x$. При этом дорога между городами (x_1, y_1) и (x_2, y_2) существует только в одном из таких двух случаев:

1. Если $x_1 = x_2 = x$, а между городами y_1 и y_2 Игреково проложена дорога. При этом длина дороги между городами (x, y_1) и (x, y_2) Декартово равно длине дороги между городами y_1 и y_2 Игреково.
2. Если $y_1 = y_2 = y$, а между городами x_1 и x_2 Иксевщины проложена дорога. При этом длина дороги между городами (x_1, y) и (x_2, y) Декартово равно длине дороги между городами x_1 и x_2 Иксово. Города разных государств между собой дорогами не соединены.

Некоторые дороги Декартовщины требуется закрыть. Ваша задача — определить, дороги какой наименьшей суммарной длины можно оставить в Декартовщине, чтобы из любого ее города все еще можно было попасть в любой другой.

Формат входных данных

Первая строка содержит натуральные числа N_x и M_x ($1 \leq N_x, M_x \leq 5 \cdot 10^4$) — количество городов и дорог в Иксово. В последующих M_x строках описаны дороги Иксово: в каждой строке по три числа, где первые два задают номера разных городов, соединенных дорогой, а третье есть длина соответствующей дороги (натуральное число, которое не превышает 10^7).

В следующей строке входного файла указаны натуральные числа N_y и M_y ($1 \leq N_y, M_y \leq 5 \cdot 10^4$) — количество городов и дорог в Игреково. Последующие M_y строк содержат описание дорог Игреково; формат данных и ограничения соответствуют описанным выше.

Формат выходных данных

Выходной файл должен содержать единственно целое число — ответ на вопрос подзадачи.

Примеры

cartesius.in	cartesius.out
3 2	117
2 1 15	
3 1 14	
3 2	
2 1 15	
3 2 15	

Задача D. Диаграммы Юнга выходят в интернет

Имя входного файла: `dfs.in`
Имя выходного файла: `dfs.out`
Ограничение по времени: 4 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.

Примеры

<code>dfs.in</code>	<code>dfs.out</code>
4	4
4 3	1 3 2 4
2 4	0
1 3	0
3 2	2
2 2	1 4
1 2	
2 1	
3 1	
2 3	
4 4	
1 2	
2 4	
3 4	
1 3	

Задача Е. Таня и пароль

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

Пока папа был на работе, маленькая девочка Таня решила поиграть с папиным паролем к секретной базе данных. Папин пароль представляет собой строку, состоящую из $n + 2$ символов. Она выписала все возможные n трёхбуквенных подстрок пароля на бумажки, по одной на каждую бумажку, а сам пароль выкинула. Каждая трёхбуквенная подстрока была выписана на бумажки столько раз, сколько она встречалась в пароле. Таким образом, в итоге у Тани оказалось n бумажек.

Потом Таня поняла, что папа расстроится, если узнает о ее игре, и решила восстановить пароль или, по крайней мере, хотя бы какую-то строку, соответствующую получившемуся набору трёхбуквенных строк. Вам предстоит помочь ей в этой непростой задаче. Известно, что папин пароль состоял из строчных и заглавных букв латинского алфавита, а также из цифр. Строчные и заглавные буквы латинского алфавита считаются различными.

Формат входных данных

В первой строке следует целое число n ($1 \leq n \leq 2 \cdot 10^5$), количество трёхбуквенных подстрок, которые получились у Тани.

Следующие n строк каждая содержат по три буквы, образующие подстроку пароля папы. Каждый символ во вводе — строчная или заглавная буква латинского алфавита или цифра.

Формат выходных данных

Если во время игры Таня что-то напутала, и строк, соответствующих данному набору подстрок, не существует, то выведите «NO».

Если же возможно восстановить строку, соответствующую данному набору подстрок, то выведите «YES», а затем любой подходящий вариант пароля.

Примеры

стандартный ввод	стандартный вывод
5 aca aba aba cab bac	YES abacaba
4 abc bCb cb1 b13	NO
7 aaa aaa aaa aaa aaa aaa aaa	YES aaaaaaaaa