

Задача А. Улиточки

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

Две улиточки Маша и Петя сейчас находятся на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до n и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). Ввиду соображений гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

Формат входных данных

В первой строке файла записаны четыре целых числа — n , m , a и h (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика).

В следующих m строках записаны пары чисел. Пара чисел (x, y) означает, что есть дорожка с лужайки x до лужайки y (из-за особенностей улиток и местности дорожки односторонние).

Ограничения: $2 \leq n \leq 10^5$, $0 \leq m \leq 10^5$, $a \neq h$

Формат выходных данных

Если существует решение, то выведите YES и на двух отдельных строчках сначала путь для Машеньки (т.к. дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

Примеры

snails.in	snails.out
3 3 1 3	YES
1 2	1 3
1 3	1 2 3
2 3	

Задача В. Максимальный поток

Имя входного файла: **flow2.in**
Имя выходного файла: **flow2.out**
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вам задан ориентированный граф G . Каждое ребро имеет некоторую пропускную способность. Найдите максимальный поток между вершинами 1 и n .

Формат входных данных

Первая строка входного файла содержит n и m — число вершин и ребер в графе ($2 \leq n \leq 500$, $1 \leq m \leq 10\,000$). Последующие строки описывают ребра. Каждое ребро задается тремя числами: начальная вершина ребра, конечная вершина ребра и пропускная способность ребра. Пропускные способности не превосходят 10^9 .

Формат выходных данных

Выведите величину максимального потока между вершинами 1 и n .

Далее для каждого ребра выведите величину потока, текущую по этому ребру.

Примеры

flow2.in	flow2.out
4 5	3
1 2 1	1
1 3 2	2
3 2 1	1
2 4 2	2
3 4 1	1
4 5	4
1 2 2	2
1 3 2	2
2 3 1	0
2 4 2	2
3 4 2	2
4 3	4
1 2 1	0
1 3 1	0
1 4 4	4

Задача С. 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

color.in	color.out
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	GBBR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

Задача D. Проблемы с Интернетом

Имя входного файла: **dfs.in**
Имя выходного файла: **dfs.out**
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Лиза и Сара раскрыли крупный заговор, и теперь вынуждены скрываться от Роскомнадзора. Держаться вместе очень рискованно, так как в случае чего повяжут их обеих, поэтому они вынуждены общаться через Интернет. Но "обычный Интернет" полностью контролируется Роскомнадзором, 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 Лизы связан с сервером 1, а компьютер Сары — с сервером n .

Роскомнадзор хочет перехватить сообщение Лизы. Для этого ему необходимо взломать такой сервер, что сообщение, посланное Лизой, по любому пути к Саре 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

dfs.in	dfs.out
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. Карточки

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

Очередной день в ЛКШ. Сегодня вечером вы не захотели пойти на интеллектуалку. Вместо этого вы придумали себе более увлекательное занятие: перед вами разложено n карточек, с обеих сторон которых написаны буквы. Перед уходом ваш друг написал какую то строчку, оставив записку рядом. И вот вы придумали задачу — а возможно ли сложить эту строчку, используя имеющиеся карты? Каждую карту можно переворачивать и менять местами с другими. Более того, некоторые карты можно вовсе не использовать. Ваша задача — ответить на поставленный вопрос.

Формат входных данных

На первой строке стандартного ввода будет задано число n ($1 \leq n \leq 100\,000$) — количество карточек перед вами. В последующих n строках будут заданы сами карточки. Каждая из этих строк содержит ровно по две строчные латинские буквы. Первая из них — символ, написанный на лицевой стороне, а вторая — символ на обратной стороне. В последней строке задана сама строка из маленьких латинских букв. Она непуста и состоит не более, чем из 100 000 символов.

Формат выходных данных

Если возможно сложить заданную строку, выведите набор чисел, равный по размеру длине строки. При этом i -е число должно быть равно по модулю номеру карточки, которая будет соответствовать i -ому символу строки. Это число должны быть положительным, если карточка взята лицевой стороной, и отрицательно, если карточка взята обратной стороной.

Примеры

стандартный ввод	стандартный вывод
3 ab ab bc bca	-1 -3 2
3 ab ab bc bcc	IMPOSSIBLE

Задача F. Часепитие

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

В одном из отделов крупной организации работает n человек. Как практически все сотрудники этой организации, они любят пить чай в перерывах между работой. При этом они достаточно дисциплинированы и делают в день ровно один перерыв, во время которого пьют чай. Для того, чтобы этот перерыв был максимально приятным, каждый из сотрудников этого отдела обязательно пьет чай одного из своих любимых сортов. В разные дни сотрудник может пить чай разных сортов. Для удобства пронумеруем сорта чая числами от 1 до m .

Недавно сотрудники отдела купили себе большой набор чайных пакетиков, который содержит a_1 пакетиков чая сорта номер 1, a_2 пакетиков чая сорта номер 2, ..., a_m пакетиков чая сорта номер m . Теперь они хотят знать, на какое максимальное число дней им может хватить купленного набора так, чтобы в каждый из дней каждому из сотрудников доставался пакетик чая одного из его любимых сортов.

Каждый сотрудник отдела пьет в день ровно одну чашку чая, которую заваривает из одного пакетика. При этом пакетики чая не завариваются повторно.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 50$). Вторая строка содержит m целых чисел a_1, \dots, a_m ($1 \leq a_i \leq 10^6$ для всех i от 1 до m).

Далее следуют n строк — i -я из этих строк описывает любимые сорта i -го сотрудника отдела и имеет следующий формат: сначала следует положительное число k_i — количество любимых сортов чая этого сотрудника, а затем идут k_i различных чисел от 1 до m — номера этих сортов.

Формат выходных данных

Выведите одно целое число — искомое максимальное количество дней.

Примеры

tea.in	tea.out
2 3	3
3 2 1	
2 1 2	
2 1 3	