

Задача 1. Размещение аппаратуры

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	6 секунд
Ограничение по памяти:	64 мегабайт

При компоновке научных отсеков «Тахмасиба» возникла проблема. Согласно требованиям безопасности, металлические шкафы с аппаратурой должны быть размещены в двух уровнях: один ряд шкафов ставится на пол, второй — точно такой же длины — подвешивается на переборку. У планетологов имеется N различных типов шкафов для нижнего ряда и M различных типов шкафов для верхнего ряда. При этом для каждого типа шкафов, кроме длины, задан вес. Естественно, что помещения на борту корабля хочется использовать максимально полно, поэтому при компоновке отсека требуется максимизировать длину блока шкафов с аппаратурой. Ваша задача — для заданной длины отсека определить самый длинный вариант компоновки для блока шкафов с аппаратурой и границы суммарного веса шкафов для предлагаемой длины. Длины верхнего и нижнего уровня должны совпадать. Шкаф каждого типа можно использовать несколько раз.

Формат входного файла

В первой строке входного файла содержатся три целых числа L , M и N : длина научного отсека в сантиметрах L , количество типов шкафов для верхнего уровня M , для верхнего ряда, количество типов шкафов для нижнего уровня N . ($0 \leq M, N \leq 1000$, $0 \leq L \leq 1000$). Вторая строка содержит M натуральных чисел — длины в сантиметрах шкафов верхнего уровня, третья строка — их веса в порядке, заданном второй строкой. Аналогично, четвёртая строка содержит N натуральных чисел — длины в сантиметрах шкафов нижнего уровня, а пятая — их веса в порядке, заданном четвёртой строкой. Далее следуют две строки с аналогичными данными для шкафов нижнего уровня. Все длины и веса задаются целыми положительными числами, не превосходящими 1000.

Формат выходного файла

Если решение существует, то в первую строку выходного файла Вы должны вывести текст **Max length N**, где N — максимальная длина проектируемого блока аппаратуры в сантиметрах. Вторая строка должна содержать текст **The weight from W_{min} to W_{max}** , где W_{min} — минимальный возможный вес, а W_{max} — максимальный.

Если существует единственный вариант веса, равный W , Вы должны вывести во второй строке текст **The weight W**.

Если из заданного набора не удаётся собрать две линии шкафов одинаковой длины, выведите в первую строку выходного файла текст **Impossible**.

Пример

input.txt	output.txt
6 4 5 7 8 9 3 1 2 3 4 11 12 4 7 6 30 12 4 12 13	Max length 6 The weight 21
7 4 3 5 4 3 1 12 13 1 12 7 2 22 100 200 1	Max length 7 The weight from 114 to 184
5 1 4 2 1 3 1	Impossible

Задача 2. Folding

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	6 секунд
Ограничение по памяти:	64 мегабайт

Программное обеспечение, использовавшееся на борту «Тахмасиба» было предоставлено в исходных текстах. Однако даже в превосходно документированном коде разобраться бывает очень сложно. Поэтому одной из задач, поставленных перед группой разработки программного обеспечения, стало написание редактора, поддерживающего следующие возможности.

Редактор работает с древовидной структурой классов некоторой программы. Каждый класс может быть вложен не более чем в один другой класс. Пользователь может просматривать эту структуру и прodelывать над ней следующие операции:

- скрыть описание некоторого класса (вместе с ним скрываются описания всех вложенных классов, вложенных в эти классы и т.д.);
- раскрыть описание какого-либо класса, при этом состояние вложенных в него классов никак не должно изменяться.

При закрытии закрытого класса и открытии открытого ничего не должно изменяться.

В рамках работы над редактором Вам поручена следующая задача: дана иерархия классов и последовательность операций в порядке их выполнения пользователем. Необходимо после каждой такой операции вывести размер видимой программы — т.е. сумму размеров всех видимых (не скрытых) классов.

Формат входного файла

В первой строке входного файла записаны два целых числа M и N — количество классов и количество операций, сделанных, пользователем. ($1 \leq M \leq 10000, 0 \leq N \leq 5000$) Далее идут две независимые части. Первая часть содержит описание иерархии классов. Каждый класс описывается в следующем формате:

```
class X (Y) {  
    X1, X2, ..., Xn;  
};
```

где X — название описываемого класса, $0 \leq Y \leq 100$ — целое число, размер класса, когда его описание раскрыто, $X1, X2 \dots Xn$ — имена вложенных в него классов. Имя класса — строка, состоящая из латинских букв, различного регистра — большие и маленькие буквы различаются, при этом имя класса не может быть длиннее 10 букв. Описание каждого класса всегда занимает три строки. Если класс не содержит вложенных классов, то вторая строка остаётся пустой.

Затем идет описание последовательности операций. Каждой операции соответствует одна строка, в которой операция выводится в следующем формате: либо $+X$, либо $-X$, что означает либо раскрытие класса с именем X , либо скрытие класса X соответственно. Изначально все классы раскрыты.

Формат выходного файла

Выходной файл должен содержать N строк. Каждая строка соответствует одной операции, заданной во входном файле. Для каждой операции нужно вывести общий размер раскрытых классов после её выполнения.

Пример

input.txt	output.txt
3 4 class A (5) { B,C; } class B(10) { } class C(10) { } -A +B -B +A	0 10 0 5

Задача 3. Перед отлётом

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	6 секунд
Ограничение по памяти:	64 мегабайт

Перед отлётом «Тахмасиба» из международного космопорта Мирза-Чарле бортинженер Иван Жилин решил выбраться в город, рассчитывая, что успеет вернуться ко времени отлёта.

Город и космопорт соединены шоссе с двумя полосами движения (по одной полосе в каждую сторону). Таким образом, для обгона попутной машины на таком шоссе необходимо на какое-то время выехать на встречную полосу. Из-за условий видимости и по другим причинам, на некоторых участках дороги такой обгон разрешен, а на некоторых других - запрещен. Это обозначается разметкой дороги: на тех участках, где выезд на встречную полосу для обгона разрешен, разделительная полоса прерывистая, а на тех, где запрещен — сплошная.

Жилин едет на машине из города в космопорт. К сожалению, оказалось, что в этот день недели дорога плотно занята тяжелыми грузовиками. Эти грузовики движутся по шоссе с постоянной скоростью и постоянным интервалом друг между другом. Интервал достаточен, чтобы легковая машина Жилина могла встроиться между любыми двумя грузовиками. Движение по встречной полосе пренебрежимо слабое, так что Иван может обгонять грузовики в любой момент, когда это разрешено разметкой дороги.

Вам дан список участков дороги, на которых обгон разрешен, скорость движения грузовиков, интервал между ними, а также динамические параметры машины Жилина (ускорение разгона и торможения и максимальная скорость). Вам необходимо определить минимальное время, за которое Жилин может доехать до космопорта, не нарушая правил движения (т.е. не находясь на встречной полосе на участках, на которых обгон запрещен).

Обгон происходит следующим образом: на момент достижения начала участка, где обгон разрешен, машина Жилина всегда имеет скорость, равную V , совпадающую со скоростью движения грузовиков. После выезда на встречную полосу машина может начать увеличивать скорость, причём величина ускорения не должна превышать A_a . На момент достижения конца участка обгона скорость машины Жилина снова должна стать равной V . Для этого он должен успеть затормозить, учитывая, что величина замедления не должна превышать по модулю A_b . Только после этого Жилин должен начать маневр возврата в свой ряд. Ускорение или торможение возможно, только когда вся машина целиком находится на разрешённом участке. На тех участках, где обгон не разрешён, машина Жилина всегда движется со скоростью колонны, и не может ни ускоряться, ни тормозить. Временем выезда на встречную полосу и встраивания обратно в колонну можно пренебречь, считая его равным нулю.

Формат входного файла

В первой строке входного файла записаны через пробел три вещественных числа V , D и L , где V — скорость движения грузовиков, м/сек, с точностью до см/сек ($0 < V \leq 500$), D — интервал между грузовиками в метрах, с точностью до сантиметров ($0 < D \leq 1000$), L — длина грузовика в метрах, с точностью до сантиметров ($0 \leq L \leq 10$). Во второй строке записаны через пробел шесть чисел l , A_a , A_b и v , где l — длина машины Жилина в метрах, с точностью до сантиметров ($0 < l < D$). A_a — ускорение, которое машина Жилина способна развивать при разгоне, в m/sec^2 , с точностью до cm/sec^2 ($0 < A_a \leq 100$); A_b — ускорение, которое машина Жилина способна развивать при торможении, в m/sec^2 , с точностью до

cm/sec^2 ($0 < A_b \leq 100$); v — максимальная скорость машины Жилина ($V \leq v \leq 500$).

В третьей строке записаны два числа: вещественное S — расстояние от города до космопорта, в метрах с точностью до сантиметров ($0 < S < 1000000$) и целое N — количество участков, на которых разрешен обгон ($0 \leq N \leq 1000$).

В следующих N строках располагается описание каждого участка. Это два числа B_i и E_i , записанные через пробел, первое из которых — расстояние от города до начала участка, а второе — расстояние от города до его конца. Расстояния даны в метрах с точностью до сантиметров.

Участки не перекрываются и не соприкасаются, отсортированы по началу участка. В момент времени 0 машина Жилина своей хвостовой частью находится в начале дороги (в точке с расстоянием 0 от города), на расстоянии D метров от хвостовой части ближайшего спереди грузовика, и движется со скоростью V . Считаем, что машина достигла конца пути, если она полностью прошла точку с координатой S .

Формат выходного файла

В выходной файл необходимо вывести одно целое число t — минимальное время в секундах, за которое Жилин может доехать до космопорта.

Пример

input.txt	output.txt
20 500 5 5 5 5 40 1000 1 0 1000	28
20 500 5 5 5 5 40 1000 2 30.33 500.85 700.77 800.88	40

Задача 4. Пасьянс

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени:
Ограничение по памяти: 64 мегабайт

Командир «Тахмасиба» Алексей Быков во время отдыха любит раскладывать пасьянс. Колода карт (не обязательно полная — что под руку попадётся) раскладывается следующим образом: первая (верхняя) карта выкладывается на стол вверх рисунком, вторая «под низ» колоды, третья снова на стол и так далее пока вся колода не будет разложена.

Как-то раз, наблюдая за этим процессом, планетолог Юрковский задумался, в каком порядке необходимо изначально расположить карты в колоде, чтобы получился заранее заданный расклад. Напишите программу, решающую эту задачу.

Формат входного файла

Во входном файле через пробел перечислена непустая последовательность карт, которая должна получиться у Быкова после раскладывания пасьянса. Каждая карта задается последовательностью символов, содержащей достоинство карты (2, 3, ... 10, J, Q, K, A) и ее масть (H, D, C, S). Все карты в заданной последовательности различны.

Формат выходного файла

В выходной файл необходимо вывести через пробел исходный порядок карт в колоде в том же формате, что и во входном файле.

Пример

<code>input.txt</code>	<code>output.txt</code>
2H 3H 4H 5H 2D 3D 4D 5D	2H 2D 3H 4D 4H 3D 5H 5D

Задача 5. Инцидент с грузовым кораблём

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	6 секунд
Ограничение по памяти:	64 мегабайт

«Тахмасиб» принял сигнал SOS. Быков тут же изменил курс корабля, и через несколько часов планетолёт оказался на месте бедствия. Оказалось, что один из кораблей, перевозивший на Луну потребовавшиеся для каких-то научных экспериментов брёвна из специально выведенных на Земле сортов дерева, столкнулся с проблемами, связанными с нарушением центровки.

После того, как ситуацию удалось стабилизировать, Генеральный Инспектор Юрковский решил разобраться в причине инцидента и вместе с бортиженером Жилиным проследовал на борт корабля. Оказалось, что технология загрузки брёвен в бункер была нарушена на Земле — брёвна просто сбрасывались в бункер. Так как длина бревна соответствует длине бункера, то оси брёвен расположены параллельно дну бункера и его боковой стенке, и при моделирования процесса на экране монитора можно обойтись проекциями распилов брёвен на переднюю стенку бункера в виде кругов. Выяснилось, что в процессе загрузки, зафиксированном бортовым компьютером, бревна падали строго по вертикали (считаем, что брёвна имеют бесконечно большой коэффициент трения). После того, как бревно коснется дна бункера или ранее упавших бревен, или бревна и стенки бункера, оно перекатывается и падает до тех пор, пока не займет устойчивое положение, то есть, пока его центр не будет проецироваться между точками его касания. При этом расположение ранее упавших бревен не меняется. Если диаметр падающего бревна меньше либо равен расстоянию между двумя соседними неподвижными бревнами, то это бревно не застрянет между ними, а будет продолжать падать.

В случае, если бревно имеет только одну точку касания, его положение будет устойчивым, если оно касается дна бункера, и неустойчивым, если оно касается другого бревна. В последнем случае гарантируется, что ситуация, когда центр падающего бревна располагается строго над центром лежащего, является невозможной.

Требуется по записи процедуры погрузки восстановить окончательное расположение брёвен в бункере.

Формат входного файла

В первой строке входного файла записаны через пробел два целых числа — S и K , где S — ширина бункера, K — количество брёвен ($1 \leq S < 1000, 1 \leq K \leq 200$). В последующих K строках приведено по два вещественных числа: R_i — радиус i -го падающего бревна ($10^6 - 4 \leq R_i \leq 100$), и расстояние L_i от его центра до левого края бункера. При этом $R_i \leq L_i \leq S - R_i$. Высота бункера считается бесконечной.

Формат выходного файла

В выходной файл нужно выдать K строк, в каждой строке должно быть записано через пробел по два вещественных числа, округленных до 7 знаков после десятичной точки — соответственно координаты X и Y всех бревен в том же порядке, в каком они перечислены во входном файле, начиная с первого. Начало координат совмещено с левым нижним краем бункера, ось OX направлена вправо.

Пример

input.txt	output.txt
12 3	5.0000000 3.0000000
3 5	2.0000000 7.0000000
2 4	9.0000000 1.0000000
1 6	

Задача 6. База на Марсе

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 6 секунд
Ограничение по памяти: 64 мегабайт

Марсианская база представляет из себя N лабораторий или жилых корпусов, соединённых дорогами. К сожалению, нумерация сооружений на базе велась по хронологическому принципу, так что начиная с какого-то момента она стала скорее мешать, чем помогать. К прибытию на «Тахмасибе» Генерального Инспектора было принято решение перенумеровать корпуса так, чтобы административный корпус имел номер 1, соединённый с ним дорогой причал для космических кораблей — номер N , а для всех остальных сооружений (которым достались бы номера в диапазоне от 2 до $N - 1$) выполнялось бы следующее требование: от каждого из сооружений должна существовать дорога к корпусу с меньшим номером и дорога к корпусу с большим номером. Ваша задача — выяснить, возможна ли такая перенумерация, и, если возможна, то построить её.

Формат входного файла

В первой строке входного файла записано число N — количество корпусов ($2 \leq N \leq 100000$). Все корпуса пронумерованы числами от 1 до N . В следующей строке записана пара целых чисел, обозначающих номера административного корпуса и космостанции — первого и последнего корпусов в новой нумерации. Затем указано общее количество дорог — неотрицательное целое число, не превосходящее 100000, а в следующих строках приводится описание дорог, по одной в строке. Каждая дорога задается двумя целыми числами, записанными через пробел — номерами корпусов, которые она соединяет. При этом между двумя различными корпусами может быть только одна дорога, и из любого корпуса в любой всегда можно пройти по дорогам.

Формат выходного файла

В выходной файл необходимо вывести последовательность старых номеров корпусов, расположенных в порядке новой нумерации. Если существует несколько решений, то вывести любое. Если перенумерация невозможна, выдать No.

Пример

input.txt	output.txt
3 1 2 2 1 2 2 3	No
4 1 2 5 1 2 1 3 2 3 3 4 4 2	1 3 4 2

Задача 7. Охота на пиявок

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 6 секунд
Ограничение по памяти: 64 мегабайт

На карте Тёплого Сырта входы в логова марсианских пиявок обозначены красными точками. Перед охотой на пиявок территорию, занятую этими логовами, требуется разбить на секторы. На карте каждый сектор должен представлять собой треугольник с вершинами, выбранными среди вершин выпуклой оболочки множества всех красных точек так, что никакие два треугольника не могут иметь общих внутренних точек и не существует точек внутри или на границе выпуклой оболочки, не принадлежащей какому-нибудь сектору. Сопоставим каждому треугольнику количество входов внутри этого треугольника и на его границе, и назовём эту величину «пиявкоопасностью» сектора.

Для того, чтобы распределение фронта работ для всех групп охотников было более или менее равномерным, требуется выбрать разбиение так, чтобы разность между секторами с наибольшей и наименьшей пиявкоопасностью была бы минимальна.

Формат входного файла

В первой строке входного файла записано одно целое число N — количество входов в логова пиявок ($3 \leq N \leq 70$). В следующих N строках записано через пробел по два числа $-10000 \leq X \leq 10000$ и $-10000 \leq Y \leq 10000$ — координаты каждой точки.

Формат выходного файла

В первую строку выходного файла необходимо вывести одно целое число — минимальную разность между максимальной и минимальной пиявкоопасностью секторов.

В следующих строках нужно записать полученное разбиение на треугольники для выпуклой оболочки. В каждой строчке записать по три целых числа — номера входов в порядке их перечисления во входном файле. Входы нумеруются с единицы. Порядок вывода секторов, а также порядок вывода вершин в каждой строке значения не имеют.

Пример

<code>input.txt</code>	<code>output.txt</code>
5	0
0 0	1 3 4
2 0	1 2 4
0 2	
2 2	
1 1	

Задача 8. Позывные

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	6 секунд
Ограничение по памяти:	64 мегабайт

При выборе позывных для научных станций, расположенных в поясе астероидов, возникла следующая проблема. Как известно, существуют слова, переставив конец и начало которых, можно снова получить осмысленное слово. В русском языке такие пары образуют, например, *пломба* и *апломб*, *цоколь* и *кольцо*, *окорок* и *рококо*; в английском — *owl* (сова) и *low* (низкий), *lyre* (лира) и *rely* (полагаться). Может оказаться так, что из одного слова можно получить несколько других различных слов. И если два слова из одной пары используются для позывных, то может произойти опасная путаница (как известно, при вызове слово повторяется многократно).

Для предотвращения подобных ситуаций Генеральный Инспектор поручил Вам написать программу, которая в заданном списке слов, выбранных различными станциями или кораблями для позывных, найдёт все множества слов, которые можно получить одно из другого описанным способом. Количество букв B в переставляемом отрезке должно быть $1 \leq B \leq L - 1$, где L — длина исходного слова.

Формат входного файла

В первой строке входного файла записано число N ($2 \leq N \leq 30000$). В следующих N строках записаны слова по одному в строке, в алфавитном порядке. Слова могут состоять из латинских строчных и прописных букв. Длина любого слова $2 \leq L \leq 100$ букв. Все слова различные. Большие и маленькие буквы различаются.

Формат выходного файла

В первую строку выходного файла нужно записать одно число K — количество найденных множеств слов, а далее в K строках нужно перечислить все найденные множества слов, по одному множеству на строке. Слова из одного множества должны быть отсортированы в лексикографическом порядке. Все множества тоже отсортированы в лексикографическом порядке по своим первым словам. Нужно выводить только множества, которые состоят более чем из одного слова.

Пример

input.txt	output.txt
14	5
German	German manGer
aaabbb	aaabbb bbbaaa
abcdef	abcdef cdefab fabcde
bbbaaa	low owl
cccb	lyre rely
cdefab	
fabcde	
low	
lyre	
manGer	
owl	
rely	
try	
zzzzz	

Задача 9. Генеральный Инспектор за работой

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	6 секунд
Ограничение по памяти:	64 мегабайт

После прибытия «Тахмасиба» на один из астероидов представители местной администрации, руководители планетологической базы, пожелали переговорить с Генеральным Инспектором, тем более, что им оказался их старый знакомый и коллега, известный планетолог Владимир Сергеевич Юрковский.

На этом астероиде несколько западных компаний развернули добычу алмазов. Колония горнодобытчиков, как и все земные поселения вне Земли, представляла собой систему корпусов, занумерованных от 1 до N , некоторые из которых соединены дорогами. Проезд по каждой дороге является платным и стоит фиксированную сумму для каждой из дорог.

Проблема состояла в том, что на астероиде открыла своё представительство крупнейшая рекламная компания, которая вознамерилась пускать по дорогам между корпусами вездеходы с нанесённой на борта рекламой. Вездеход должен был следовать от ангара, расположенного в корпусе u , до представительства рекламной компании v , при этом обязательно проехать мимо корпуса правления корпораций, расположенного под номером s — именно там находится наибольшее число потенциальных покупателей. При этом, чтобы реклама не была назойливой и не вызвала обратный эффект, вездеход не должен был дважды оказываться ни у одного из корпусов. Компании требуется найти минимальную сумму, которую необходимо затратить на то, чтобы проехать маршрутом с подобными свойствами.

И до тех пор, пока сумма не будет рассчитана, компания отказывается платить налог. «Позакрывать бы их всех» — заметил Юрковский, но задачу решил. Посмотрим, справитесь ли с ней вы.

Формат входного файла

В первой строке входного файла записано два целых числа N и M — количество корпусов и дорог на астероиде соответственно ($3 \leq N \leq 5000, 0 \leq M \leq 10000$). В следующих M строках записано по три целых числа, первые два из которых — номера корпусов, которые соединяет дорога, а третье — стоимость проезда по этой дороге. Между двумя корпусами может быть больше одной дороги. Дорога может начинаться и заканчиваться около одного и того же корпуса.

В последней строке записано три числа u , v и s — номера ангара, корпуса рекламной компании и корпуса правления корпораций соответственно. Совпадающих корпусов с различными номерами нет. Плата на проезд по каждой дороге неотрицательна и не превосходит 10000.

Формат выходного файла

В первую строку выходного файла необходимо вывести одно целое число — требуемую минимальную сумму. В следующей строке должен быть выведен искомый путь — последовательность номеров корпусов, начинающаяся с номера u , заканчивающаяся номером v и содержащая номер s . Если таких путей несколько, то вывести любой. Если пути, удовлетворяющего условиям задачи, не существует, то необходимо вывести **Impossible!**.

Пример

input.txt	output.txt
4 4 1 2 1 2 3 1 3 4 1 4 1 1 1 4 2	3 1 2 3 4
4 3 1 2 1 2 4 1 4 1 1 1 4 3	Impossible!

Задача 10. Музей

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	6 секунд
Ограничение по памяти:	64 мегабайт

На одном из спутников Сатурна Владимир Юрковский и Григорий Дауге случайно наткнулись на странное сооружение. Оказалось, что это — древний музей, созданный внеземной цивилизацией. Рядом с музеем были расположены два лабиринта. После расшифровки найденных надписей выяснилось, что механизм открытия двери музея сработает, если электронный ключ будет нажат одновременно в двух лабиринтах в одинаково расположенных на плане комнатах.

Лабиринт состоит из N комнат, комнаты пронумерованы от 1 до N .

Все комнаты абсолютно одинаковы, в них никак нельзя оставить отметки или вещи. На устройстве перемещения из комнаты в комнату есть набор из C попарно разноцветных кнопок. Комната, в которую перемещается исследователь вместе с устройством при нажатии кнопки, однозначно определяется цветом нажатой кнопки и комнатой, в которой он находился в момент нажатия на кнопку. Трудность заключается в том, что при входе в лабиринт не известно, в какую комнату попадаешь, причем при каждом новом входе можно попасть в любую другую комнату, а связь с внешним миром внутри лабиринта не работает.

Ученые нашли планы лабиринтов с указанием перемещений из комнаты в комнату при нажатии цветной кнопки. Оказалось, что два лабиринта одинаковы. После ряда экспериментов было замечено, что по задумке создателей или от долгого бездействия устройство контроля нажатий кнопок не выдерживает больше N^3 нажатий в обоих лабиринтах сразу.

Можете ли вы им помочь и найти такую последовательность нажатий кнопок, которая гарантирует, что два исследователя, войдя в разные лабиринты и не зная своих начальных комнат, после перемещений, вызванных вашей последовательностью, окажутся в одинаковых по номеру комнатах и смогут активировать ключ?

Формат входного файла

В первой строке входного файла указано количество комнат $1 \leq N \leq 100$ и количество различных цветов на кнопках $C (1 \leq C \leq 100)$. Далее тройками целых чисел задаются перемещения — номер комнаты отправления, номер комнаты прибытия, номер цвета нажимаемой кнопки. (Будем считать, что цвета пронумерованы от 1 до C).

Формат выходного файла

В первую строку выходного файла необходимо выдать длину последовательности перемещений, а во вторую — саму последовательность цветов нажимаемых кнопок. Номера цветов в этой последовательности выдаются через пробел.

Если ответов, удовлетворяющих условиям, несколько, то можно выдать любой.

Пример

input.txt	output.txt
3 2	2
1 2 2	2 1
1 3 1	
2 3 1	
2 1 2	
3 1 2	
3 3 1	