

Задача А. Простая задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

*Человек полетит, опираясь не на силу
своих мускулов, а на силу своего разума.*

Николай Жуковский

Около десятка лет группа учёных СГАУ работала над секретным правительственным проектом, целью которого был полёт русских космонавтов на Марс. Полёт совершенно случайно должен был состояться 12-го апреля 2011 года, но сроки уже подходили, а проект так и не был проработан полностью, хотя оставалось решить не так уж много вопросов.

Главной проблемой был вид топлива, который будет использован для приведения космического корабля в движение. Профессор X (очень известный сотрудник СГАУ) уже давно настаивал на изобретённом им «водородном черпаке», а его коллега профессор Y предлагал для верности использовать пару «керосин-кислород». Чтобы полёт не сорвался из-за этого спора, как это было с полётом на Луну, коллеги решили сыграть в игру, исход которой и должен был решить топливный вопрос.

Чтобы быть честными с собой, они поймали в коридоре первого попавшегося студента и попросили его сочинить им правила игры, в которую ещё никто не играл. Тот, немного подумав, высыпал из кармана на стоявшую неподалёку парту довольно большую кучу печенек и предложил немного упрощённую схему известной игры Ним.

Профессора ходят по очереди, первым ходит профессор X. За свой ход каждый профессор может взять из кучки и съесть количество печенек, равное некоторой целой неотрицательной (в том числе нулевой) степени какого-нибудь простого числа (натуральное число называется простым, если имеет ровно два различных натуральных делителя). Съевший последнюю печенку выиграл.

Профессор X взглянул на кучу и мгновенно подсчитал количество печенек в ней. Он занимался математикой достаточно долго, чтобы не испытывать трудностей с устным счётом. Ему известно, что профессор Y очень умен и будет играть оптимально. Теперь и он хотел бы оценить свои шансы и придумать оптимальную для себя стратегию.

Формат входных данных

В единственной строке входного файла записано единственное целое число n ($1 \leq n \leq 10^9$) — количество печенек в куче.

Формат выходных данных

В первой строке выходного файла должна быть записана строка «X WINS» без кавычек, если при оптимальной игре обоих игроков профессор X выиграет, или строка «Y WINS» без кавычек, если выиграет профессор Y. Если выигрывает профессор X, то во второй строке должно быть записано единственное целое число — количество печенек, которое нужно первым ходом взять профессору X, чтобы выиграть при последующей оптимальной игре. Если ответов несколько, можно вывести любой из них.

Примеры

<code>input.txt</code>	<code>output.txt</code>
10	X WINS 4
12	Y WINS

Задача В. Кулинарная задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

Через тернии к звёздам.

Сенека

Профессор У очень расстроился, програв профессору Х в игру с печеньками, и захотел взять реванш. Профессор Х согласился.

— Но где же мы возьмем печеньки? — спросил профессор Х.

— Не беда, — ответил профессор У, — мы их приготовим.

Сказано-сделано. Два профессора пошли в магазин и купили несколько килограммов теста. Придя в лабораторию, они раскатали тесто по столу и стали думать, какой же формы испечь печеньки.

В лаборатории нашлось 2 вида формочек для печенек: треугольная (подставка под призму) со сторонами a , b и c и круглая (оправа от линзы) с радиусом r . После того как печеньки были приготовлены, профессор У предложил профессору Х занимательную задачу: определит, пролезет ли круглая печенье в треугольную формочку и наоборот, пролезет ли треугольная печенье в круглую формочку. Печеньки были очень тонкими, и можно было пытаться протолкнуть их в формочки, как угодно поворачивая в пространстве.

Профессор Х не был готов к такой задаче — еще бы, ведь он всю ночь готовился к игре, придуманной студентом! Поэтому он попросил Вас ответить на вопрос профессора У.

Формат входных данных

В первой строке входного файла содержится 4 целых числа через пробел: a , b , c и r ($1 \leq a, b, c, r \leq 10000$) — длины сторон треугольной формочки и радиус круглой формочки.

Формат выходных данных

В первой строке выходного файла выведите «Circle gets into the triangle» или «Circle doesn't get into the triangle», в зависимости от того, лезет ли круглая печенье в треугольную формочку или нет.

Во второй строке выходного файла аналогичным образом напишите «Triangle gets into the circle» или «Triangle doesn't get into the circle», в зависимости от случая.

Примеры

input.txt	output.txt
1 1 1 10	Circle doesn't get into the triangle Triangle gets into the circle
10 10 10 1	Circle gets into the triangle Triangle doesn't get into the circle

Задача С. Магическая задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

Летать и строить, строить и летать.

Сергей Королёв

Космический корабль, построенный для покорения красной планеты, назывался «Запад-1». Возможно, это было связано с тем, что в России стало больше западных ценностей, чем восточных, а возможно, с тем, что бортовые компьютеры были привезены с запада. Корабль должен был лететь по строго заданной на земле программе, если не случится ничего форс-мажорного, и управляться исключительно этими бортовыми компьютерами. Разумеется, была предусмотрена и возможность космонавтам взять управление на себя, однако по общей договорённости учёных, сначала космонавт должен был пройти проверку на адекватность. Был расчёт на то, что космос с его излучениями и марсианами может негативно повлиять на психическое состояние космонавта, и тот захочет необоснованно взять на себя управление.

Таким образом, чтобы получить возможность рулить космическим кораблём, космонавт, как и Юрий Гагарин в своё время, должен будет решить несложную логическую задачку. Эта задачка была придумана лично профессором Х. Он сделал её посложнее, чтобы у космонавтов отпало желание вручную управлять кораблём, если только не произойдёт что-то действительно серьёзное. Компьютер случайно выдавал космонавту натуральное число n . Тот должен был на листочке быстро построить квадрат размером $n \times n$ клеток и вписать в клетки натуральные числа от 1 до n^2 так, чтобы суммы чисел по столбцам, строкам и диагоналям квадрата были равны одному и тому же числу m . Далее, чтобы не тратить время на перерисовывание всего квадрата, он должен был ответить компьютеру получившееся число m . Теперь профессор Х сам хотел бы узнать решение этой магической задачи, потому что ему надо написать программу, которая бы проверяла решение космонавта.

Формат входных данных

В единственной строке входного файла записано единственное целое число n ($1 \leq n \leq 10^3$) — размер квадрата, который нужно построить.

Формат выходных данных

В единственной строке выходного файла должно содержаться единственное целое число — получившаяся сумма чисел, одинаковая в столбцах, строках и диагоналях квадрата. Если не существует способа построить указанный квадрат, то выходной файл должен содержать слово «FAIL» без кавычек.

Примеры

<code>input.txt</code>	<code>output.txt</code>
3	15
5	65

Задача D. Боевая задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

*Чтобы стать крылатым, нужно
стремление к полёту.*

Юрий Гагарин

Космический корабль «Запад-1» был оснащён лазерной пушкой. Разумеется, вся команда учёных, работавшая над проектом полёта на Марс, была уверена, что космос — самое мирное место на свете, никаких злобных марсиан, сбивающих зонды, не существует, НЛО — просто старая разработка советских учёных, и вообще была против оружия в космосе. Но случаи бывают разные, так что рисковать не стали.

Перед полётом нужно было проверить программу, которой был оснащён модуль наведения. Модулю на вход подавались координаты и радиус некоторого сферического звездолёта в вакууме. Координаты были в некоторой универсальной трёхмерной системе, не привязанной к чему-то конкретному. Модуль наведения выдавал координаты некоторой точки, в которую следовало стрелять. Во время выстрела лазерная пушка образует луч между кораблём (его для простоты можно считать точкой) и указанной точкой. Луч начинается в точке дислокации корабля и распространяется в одну сторону приблизительно до бесконечности (во всяком случае, модуль наведения рассчитывает на это). Теперь нужно написать программу, проверяющую работу модуля наведения. Вражеский сферический звездолёт в вакууме считается поражённым, если лазерный луч пересёк его или хотя бы коснулся.

Формат входных данных

В первой строке входного файла записаны 3 целых числа через пробел: x_s, y_s и z_s ($-10^3 \leq x_s, y_s, z_s \leq 10^3$) — координаты космического корабля «Запад-1» (в частности, лазерной пушки). Во второй строке записаны 4 целых числа через пробел: x_e, y_e, z_e и r ($-10^3 \leq x_e, y_e, z_e \leq 10^3, 1 \leq r \leq 10^3$) — координаты вражеского сферического звездолёта в вакууме и его радиус. В третьей строке записаны 3 целых числа через пробел: x_t, y_t и z_t ($-10^3 \leq x_t, y_t, z_t \leq 10^3$) — координаты цели, выданной модулем наведения. Гарантируется, что «Запад-1» не находится внутри противника и даже не касается его, а также что модуль наведения никогда не стреляет сам в себя.

Формат выходных данных

В первой строке выходного файла должно быть записано «HIT» без кавычек, если цель поражена, как это описано в условии задачи. Иначе в первой строке выходного файла должно быть записано «MISS» без кавычек. В любом случае, во второй строке должно быть записано единственное вещественное число — расстояние от центра вражеского звездолёта до выпущенного луча. Абсолютная или относительная погрешность ответа не должна превышать 10^{-6} .

Примеры

input.txt	output.txt
0 0 0 100 100 100 10 1 1 1	HIT 0.0000000000000000
0 13 9 5 13 -1 4 16 13 -3	MISS 5.0000000000000000

Задача Е. Инновационная задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

*Космонавтика имеет безграничное
будущее, и ее перспективы беспредельны,
как сама Вселенная.*

Сергей Королёв

Поскольку постройка космического корабля для полёта на Марс контролировалась вышестоящими инстанциями, в его устройстве не обошлось без нанотехнологий и прочих инноваций. Корабль был снабжён n ремонтными нанороботами размером с молекулу. Эти роботы находились во всех внутренних деталях корабля и невидимым образом чинили любые поломки. Казалось, будто поломка корабля затягивается сама собой, словно рана на теле живого существа.

На деле это работало вполне научно. Каждый из n нанороботов мог за одну секунду либо починить одно наноповреждение (очень маленькое неделимое повреждение), либо построить из подручных средств копию себя (реплицироваться). Чтобы правильно распределить работу, нанороботы связывались с бортовым компьютером для координации действий. Но пока что необходимой программы на нём не было, и роботы занимались тем, что им приходило в голову.

Профессор Х решил всё же написать такую программу, потому что начальство очень хотело, чтобы нанороботы непременно использовались. Суть программы в том, чтобы давать приказы отдельным роботам строить себе подобных или чинить корабль, в зависимости от числа имеющихся наноповреждений. Это число оценивается специальными датчиками жизнеобеспечения и передаётся программе. Естественно, нужно починить все повреждения как можно быстрее. Отдельной проблемой является то, что количество нанороботов и наноповреждений может быть очень большим.

Формат входных данных

В единственной строке входного файла содержится два целых числа через пробел: n и m ($0 \leq n, m \leq 10^{100}$) — количество нанороботов и наноповреждений соответственно.

Формат выходных данных

В первой строке выходного файла должно быть записано единственное целое число t — минимальное количество секунд, необходимых, чтобы починить корабль. Далее во второй строке должны быть записаны t целых чисел — команды нанороботам. На i -й позиции должно быть записано количество нанороботов, которым следует заниматься репликацией в i -ю секунду. Если ответов несколько, можно вывести любой из них. Если все наноповреждения починить невозможно, то в единственной строке выходного файла должно содержаться слово «IMPOSSIBLE» без кавычек.

Примеры

<code>input.txt</code>	<code>output.txt</code>
10 30	3 0 10 0
15 70	4 10 20 0 0

Задача F. Сложная задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

*Меня ужасает вечное безмолвие этих
пространств.*

Блез Паскаль

Профессор X был обеспокоен состоянием энергогенераторов космического корабля «Запад-1». Затея питаться от энергии двигателей ему не очень нравилась, потому что до Марса путь совсем неблизкий, и может случиться всё, что угодно. Что, если на обратном пути топливо кончится? Космонавтам же придётся лететь всю дорогу до Земли без света, а это очень грустно и тоскливо: ни телевизор посмотреть, ни почитать книжку, ни послушать музыку.

Чтобы решить эту сложную проблему, профессор решил запитать всё электрооборудование космолёта от независимой системы генераторов, закупленных с запада (отечественные были слишком тяжёлые, как и всё отечественное). В документации к этим генераторам было указано, что n генераторов за m минут генерируют ровно k Джоулей электроэнергии. Профессор знает, что для правильной работы всех бортовых устройств нужно, чтобы за p минут было получено хотя бы q Джоулей энергии. Теперь он хочет узнать наименьшее количество генераторов, которое ему нужно смонтировать в космический корабль, чтобы это условие выполнялось.

Формат входных данных

В единственной строке входного файла содержится пять целых чисел через пробел: n, m, k, p, q ($1 \leq n, m, k, p, q \leq 10^6$) — величины, описанные в условии задачи.

Формат выходных данных

В единственной строке выходного файла должно быть записано единственное целое число — наименьшее количество генераторов, которое нужно установить на космический корабль.

Примеры

input.txt	output.txt
2 2 2 1 3	6
1 2 3 4 5	1

Задача G. Территориальная задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

Мы всегда будем штурмовать вселенную.

Алексей Леонов

Пока учёные готовили полёт на Марс, правительства уже начали делить территории на красной планете. Особенно русских исследователей волновала одна богатая на интересные места локация размерами $n \times m$ километров, разделённая на квадраты со стороной в 1 километр. Некоторые квадраты этой локации были очень важны российскому правительству, потому что в них по предположениям некоторых учёных могли быть нефтяные месторождения, залежи урана, секретные города марсиан и прочие интересные вещи. Эти квадраты при разделе территории непременно должны были отойти Российской Федерации.

Однако, мировое сообщество поставило условие: российское правительство может заявить за собой только некоторый прямоугольник, охватывающий некоторое количество квадратов, причём это не может быть вся локация целиком. Более того, это возможно только при условии, что в течение часа дипломаты смогут ответить на вопрос, сколько всего существует таких прямоугольных территорий, включающих все интересные квадраты. Теперь один важный человек из правительства позвонил на сотовый профессору X, чтобы тот помог ему ответить на этот вопрос и не дал отечественным дипломатам опозориться перед мировым сообществом.

Формат входных данных

В первой строке входного файла содержатся три целых числа через пробел: n , m и k ($1 \leq n, m \leq 100, 1 \leq k \leq n \cdot m$) — размеры локации и количество интересных квадратов на ней. Далее в следующих k строках перечислены координаты этих квадратов. В i -й из этих строк содержатся два целых числа через пробел: x_i и y_i ($1 \leq x_i \leq n, 1 \leq y_i \leq m$) — номер строки и столбца, в которых располагается i -й интересный квадрат.

Формат выходных данных

В единственной строке выходного файла должно содержаться единственное целое число — количество различных прямоугольных территорий, описанных в условии задачи.

Примеры

input.txt	output.txt
5 5 3 2 3 3 4 4 3	23
3 7 3 1 3 2 4 3 3	11

Задача Н. Спичечная задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

Порядок освобождает мысль.

Сергей Королёв

Профессор Х очень много курил. Он уже подумывал избавиться от этой вредной привычки, но никак не находил времени и желания. А с этим полётом на Марс он от волнения стал выкуривать по две пачки сигарет в сутки. При этом профессор по старинке пользовался спичками вместо зажигалки: он любил запах сгоревшей серы, ему казалось, это придаёт некоторый особый шарм процессу курения.

В последнее время спички стали кончаться почти так же быстро, как и сигареты. При этом профессор Х очень рассеянный, поэтому постоянно забывает, в каком кармане у него лежит коробок, и лезет в один из двух карманов наугад. Кармана всего два, но всё равно неприятно каждый раз залезать в карман и обнаруживать его пустым. Чтобы этого избежать, как-то утром профессор купил два новых коробка по n спичек в каждом и положил по коробку в каждый карман. Теперь возможность ошибиться карманом на время исключалась.

Тем не менее, не прошло и пары дней, как в самый ответственный момент профессор вытащил из кармана пустой коробок. Досада сразу же прошла, когда профессор задумался, сколько в среднем спичек может быть сейчас в коробке, который лежит в другом кармане.

Формат входных данных

В единственной строке входного файла содержится единственное целое число n ($1 \leq n \leq 30$) — изначальное количество спичек в каждом из коробков.

Формат выходных данных

В единственной строке выходного файла должно быть записано единственное вещественное число — среднее количество спичек, оставшееся во втором коробке профессора. Абсолютная или относительная погрешность ответа не должна превышать 10^{-6} .

Примеры

<code>input.txt</code>	<code>output.txt</code>
2	0.8750000000000000
3	1.1875000000000000

Замечание

Министерство здравоохранения и социального развития Российской Федерации предупреждает: курение вредит вашему здоровью.

Задача I. Детская задача

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	128 мегабайт

*Земля — это колыбель разума, но нельзя
вечно жить в колыбели.*

Константин Циолковский

Полным ходом шла подготовка космонавтов, которым суждено покорять красную планету. Профессору Х предложили выбрать среди уже подготовленных и прошедших физический отбор космонавтов тех, кто покажется ему наиболее достойным. Профессор решил, что раз все космонавты проходят по физическим показателям, надо бы отобрать тех из них, кто лучше соображает.

Профессор, не долго думая, набросал на листочке несколько числовых ребусов, которые обычно дают решать детям на уроках математики, и хотел предложить их космонавтам. Но он столкнулся с проблемой, что сам не мог с ходу найти решения некоторых из них. Для ускорения процесса профессор Х решил написать программу, которая будет решить ребусы за него.

Все ребусы профессора представляли собой задачи вида « $a+b=c$ », где a , b и c — некоторые слова. Каждая буква слова может быть заменена на какую-либо цифру, причём в рамках одной задачи одна и та же буква обозначает одну и ту же цифру, а разные буквы — несовпадающие цифры. Решить задачу — значит подобрать такую замену всех букв на цифры, которое удовлетворяло бы названному условию и обеспечивало заданное равенство. Разумеется, найти нужно все решения. Получившиеся в результате замены числа могут содержать ведущие нули.

Формат входных данных

В единственной строке входного файла располагается строка вида « $a+b=c$ » — числовой ребус. Здесь a , b и c — непустые слова, являющиеся компонентами ребуса. Длина каждого из них не превышает 15 символов, все они состоят из больших букв латинского алфавита.

Формат выходных данных

В первой строке выходного файла должно содержаться единственное целое число n — количество решений указанного ребуса. Гарантируется, что число решений каждого ребуса не превышает 10^3 . Далее в n строках должны быть записаны решения. В i -й из них должна быть записана строка, получившаяся из строки « $a+b=c$ » без кавычек путём замены букв на цифры. Решения можно выводить в любом порядке.

Примеры

input.txt	output.txt
ONE+ONE=TWO	18 065+065=130 085+085=170 206+206=412 216+216=432 231+231=462 236+236=472 271+271=542 281+281=562 286+286=572 291+291=582 407+407=814 417+417=834 427+427=854 432+432=864 452+452=904 457+457=914 467+467=934 482+482=964
VOLVO+FIAT=MOTOR	10 15615+9743=25358 15715+9643=25358 36736+9825=46561 36836+9725=46561 46346+9821=56167 46846+9321=56167 71571+9642=81213 71671+9542=81213 72472+9651=82123 72672+9451=82123

Задача J. Кольцевая задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

Нас ждут бездны открытий и мудрости.

Константин Циолковский

Профессор X заботился о безопасности в любом вопросе. Поэтому при подготовке полета на Марс он также поднял тему безопасности. В частности, его внимание привлекли скафандры, которые космонавты будут надевать для выхода на поверхность планеты.

По распоряжению профессора X каждый скафандр должен быть экипирован специальной системой, которая в случае непредвиденных обстоятельств послужит хорошей страховкой космонавту. Эта система состоит из наногенератора плазменных колец, вскрывателя, закрывателя и соединителя. В случае опасности космонавт может нажать специальную кнопку на своем скафандре, и система активируется.

Работает эта система следующим образом: сначала наногенератор создаст n цепей из сверхпрочных плазменных колец, которые в дальнейшем соединятся в единую цепь и пристыкуются к кораблю «Запад-1». Чтобы соединить 2 цепи между собой, должен сначала активироваться вскрыватель, который вскрывает какое-нибудь кольцо на какой-либо цепи и отсоединяет его от этой цепи. Затем в дело вступает соединитель, который притягивает два конца каких-либо цепей в только что открытое кольцо. А потом закрыватель закрывает вскрытое кольцо.

Вся соль этой системы в том, что количество колец в цепях, созданных наногенератором, может быть разным для разных цепей. Поэтому профессор X попросил Вас написать для системы безопасности еще один модуль, который будет вскрывать и закрывать наименьшее число колец, ведь на вскрытие и закрытие тратится очень много энергии.

Формат входных данных

В единственной строке входного файла содержится единственное целое число n ($1 \leq n \leq 10^5$) — количество цепей, сгенерированных наногенератором.

В следующей строке через пробел перечислены n целых чисел a_i ($1 \leq a_i \leq 10^9$) — количество колец в i -й цепи.

Формат выходных данных

В единственной строке выходного файла должно быть записано единственное целое число — наименьшее количество колец, которые надо вскрыть и закрыть для того, чтобы можно было образовать единую цепь.

Примеры

<code>input.txt</code>	<code>output.txt</code>
5 3 3 3 3 3	3
3 1 6666 100500	1

Задача К. Системная задача

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	128 мегабайт

*12 апреля произошло смещение эпох.
Позавтракали люди в одной эпохе, а
обедали уже в другой.*

Виталий Севастьянов

Уже ближе к концу приготовлений к полёту на Марс было решено переустановить операционную систему на бортовых компьютерах космического корабля «Запад-1». Дело в том, что на нём стояла операционная система Doors, которая предназначалась для домашних компьютеров и поэтому была сделана не на совесть. Центр управления полётами СГАУ уже потерял два спутника: один упал и не поднялся, а другой и вовсе завис на орбите вопреки всяким физическим законам. Предположительно это связано именно с операционной системой.

На компьютеры было решено поставить операционную систему Lin OS X, но систему Doors тоже было решено оставить для надёжности, к тому же за неё уже были уплачены немалые деньги. Чтобы система Lin OS X работала лучше, было решено удалить всё ПО, которое установлено в системе Doors. Однако это оказалось не так-то просто, потому что для удаления одной программы было необходимо наличие другой программы, а то и не одной. Это была особенность системы Doors.

Профессор X с коллегами долго мучился над удалением программ, ведь работа системного администратора не была для них родной. В конце концов он решил подойти к проблеме с научной точки зрения. Он занумеровал программы, провёл тщательное исследование зависимостей и для каждой программы установил набор программ необходимый, чтобы её удалить. Осталось только определить, в каком порядке нужно удалять программы, чтобы без сожаления удалить их все. Если ничего не выйдет, оставался вариант с форматированием жёсткого диска, но у профессора X на бортовом компьютере хранились фотографии с последней поездки в Санкт-Петербург, и он не хотел их потерять.

Формат входных данных

В первой строке входного файла содержится единственное целое число n ($1 \leq n \leq 10^3$) — количество установленных программ. Далее в n строках содержится описание зависимостей между программами. В i -й строке содержится сначала целое число m_i ($0 \leq m_i \leq n - 1$) — количество программ, которое должно быть установлено, чтобы успешно удалить программу с номером i . Далее через пробел записаны m_i целых чисел p_{ij} ($1 \leq p_{ij} \leq n$) — номера программ, которые должны быть установлены, чтобы можно было удалить программу с номером i . Подразумевается, что для удаления программы она сама должна быть установлена. Это не пишется во входных данных.

Формат выходных данных

В единственной строке выходного файла должны быть записаны n целых чисел через пробел — номера программ в том порядке, в котором их следует удалять, чтобы избежать проблем с зависимостями. Если существует несколько вариантов ответа, можно вывести любой из них. Если придётся форматировать жёсткий диск, следует вывести единственное целое число -1 .

Примеры

input.txt	output.txt
3 2 2 3 1 3 0	1 2 3
8 2 2 3 2 4 5 2 4 7 0 1 6 0 1 8 0	1 2 3 4 5 6 7 8

Примеры

input.txt	output.txt
0 1	TRUE FALSE
99 100	TRUE TRUE

Задача М. Антинаучная задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 128 мегабайт

*Главная сила в человеке — это сила
духа...*

Юрий Гагарин

Обычный полёт на Марс с использованием топливных двигателей занял бы несколько лет. Понятно, что учёным не хотелось бы ждать так долго, поэтому у них был план. Не так давно астрономами СГАУ был открыт принципиально новый способ перемещения, основанный на пространственных разломах. В определённых областях необъятной Вселенной имеются так называемые провалы в пространстве, позволяющие вопреки законам теории относительности перемещаться на огромные расстояния за пренебрежимо малое время. Такие области учёные называли кротовыми норами. Две кротовые норы могут быть не соединены вообще, соединены гиперпространственным переходом, либо соединены нуль-переходом, причём такие соединения могут быть односторонними. В любом случае, добравшись до одной из кротовых нор, космический корабль сможет перемещаться между ними, затрачивая малое количество энергии. Если из одной норы в другую ведёт гиперпространственный переход, то переместиться между ними в заданном направлении космический корабль сможет за один античас. На корабле это перемещение действительно займёт примерно час. Если же из одной кротовой норы в другую ведёт нуль-переход, то корабль и вовсе может переместиться по нему без временных затрат: происходит мгновенный обмен участками пространства с другой норой.

Всего учёным известно положение n кротовых нор и связи между ними. Все норы занумерованы. По плану космический корабль «Запад-1» доберётся до кротовой норы номер 1, которая ближе всего находится к Земле, а дальше, перемещаясь по ним, будет стремиться попасть в нору номер n , которая находится ближе всего к Марсу. Теперь учёные хотели бы проложить маршрут перемещения между этими норами, чтобы время, потраченное космонавтами на перемещение, было минимальным. При этом учёные считают маршрут заведомо плохим, если корабль посещает одну и ту же нору дважды.

Формат входных данных

В первой строке входного файла содержится два целых числа через пробел: n и m ($1 \leq n \leq 10^5, 0 \leq m \leq 10^5$) — количество кротовых нор и количество известных связей между ними. В последующих m строках содержится по 3 целых числа через пробел: a_i, b_i, t_i — номера кротовых нор, соединённых переходом, и тип перехода (0 обозначает нуль-переход, 1 — гиперпереход). Гарантируется, что переход ни из какой норы не ведёт в неё саму, кроме того, из одной норы в другую напрямую может вести только один переход.

Формат выходных данных

В первой строке выходного файла должны быть записаны 2 целых числа через пробел — количество античасов, которое будет затрачено космонавтами на путешествие, и количество кротовых нор k в маршруте корабля. Во второй строке должно быть записано k целых чисел — номера кротовых нор, которые должен посетить корабль. Если решений несколько, можно вывести любое из них. Если не существует способа добраться до нужной кротовой норы через переходы между ними, то в единственной строке выходного файла должно быть записано единственное слово «IMPOSSIBLE» без кавычек.

Примеры

input.txt	output.txt
9 11 1 2 1 1 3 1 2 4 1 2 5 1 3 5 1 3 6 1 4 9 1 6 9 1 5 7 0 7 8 0 8 9 0	2 6 1 3 5 7 8 9
15 18 1 2 0 1 3 0 2 3 1 2 5 1 2 4 1 3 4 1 3 8 1 4 5 0 4 6 0 4 7 0 4 8 0 9 13 1 10 13 1 10 11 1 11 14 1 12 14 1 13 14 0 14 15 0	IMPOSSIBLE