

Задача Е. Билеты

Имя входного файла: e.in
Имя выходного файла: e.out
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

В одной театральной кассе есть в продаже билеты любой стоимости, выражающейся натуральным числом. При покупке билетов по цене за билет от A до B рублей включительно нужно дополнительно оплатить сервисный сбор в размере C процентов от номинальной стоимости билетов (сервисный сбор не обязательно выражается целым числом рублей, но всегда выражается целым числом копеек). При покупке билетов стоимостью менее A рублей за билет, а также более B рублей за билет, сервисный сбор не берется.

У вас есть X рублей и вам нужно K билетов одинаковой цены (цена обязательно должна выражаться натуральным числом рублей). Билеты какого самого дорогого номинала вы можете себе позволить?

Формат входных данных

Вводятся целые A, B, C, X, K ($1 \leq A \leq B \leq 10^9$, $0 \leq C \leq 1000$, $0 \leq X \leq 10^9$, $1 \leq K \leq 100000$).

Формат выходных данных

Если на имеющиеся деньги невозможно приобрести ни одного билета, выведите 0. Иначе выведите натуральное число – номинальную стоимость приобретённых билетов.

Примеры

e.in	e.out
1 10 0 5 5	1
10 100 50 50 5	9
10 100 50 100 5	13

Частичные ограничения

Первая группа состоит из тестов, в которых все числа не превосходят 100 и $X < B$.
Вторая группа состоит из тестов, в которых все числа не превосходят 10^5 .

Задача F. Петино путешествие

Имя входного файла: f.in
Имя выходного файла: f.out
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Завтра Петя уезжает в кругосветное путешествие, в процессе которого собирается посетить N разных городов. Вспомнив о старинной традиции бросать монетки в фонтаны для того, чтобы когда-нибудь вернуться в это место, он решил запастись монетами заранее. Поскольку это всего лишь традиция, подумал Петя, то с него хватит оставить в каждом городе по одной копеечной монете – зачем тратиться зря?

К сожалению, копеечные монеты – достаточно редкая вещь. В частности, таковых у Пети не нашлось. Купюр и монет всех остальных достоинств у него с избытком.

С этими мыслями Петя решил прогуляться до продуктового магазина – купить в дорогу немного еды. Из всего ассортимента ему подходило M видов товара (количество товаров каждого вида неограниченно), стоимость i -го равна a_i рублей b_i копеек. И тут его осенило. Если покупать товары в правильной последовательности, то он довольно быстро сможет скопить так нужные ему N копеечных монет!

Процесс покупки в магазине устроен следующим образом. Петя может заказать любой набор из подходящих ему товаров (каждого товара Петя может взять сколько угодно единиц). После чего он платит за них и получает сдачу минимальным числом купюр и монет (любых монет и купюр в кассе

также с избытком). Это означает, например, что если ему должны сдать 11 рублей и 98 копеек сдачи, то он получит купюру в 10 рублей, монеты в 1 рубль, 50 копеек, 4 монеты в 10 копеек, одну монету в 5 копеек и три копеечных монеты. При этом он волен вносить любую сумму (лишь бы она была не меньше требуемой для оплаты) и платить любым набором купюр и монет, имеющихся у него в распоряжении.

После этого Петя может ещё раз подойти к кассе, сделать заказ, расплатиться имеющимися наличными (можно использовать и полученные до этого со сдачей) и так далее сколько угодно раз.

Петя хочет потратить в этом магазине как можно меньше денег. Помогите ему найти оптимальный способ обретения не менее N копеечных монет с минимальными затратами.

Комментарий для нероссийских участников олимпиады.

В России используются монеты и купюры достоинством 1, 5, 10, 50 копеек и 1, 2, 5, 10, 50, 100, 500, 1000 и 5000 рублей. 1 рубль равен 100 копейкам.

Формат входных данных

Сначала вводятся целые числа N и M ($0 \leq N \leq 10^8$, $0 \leq M \leq 100$) — количество городов, которые собирается посетить Петя, и количество подходящих ему видов товара. Далее идут M пар чисел a_i, b_i , обозначающих стоимость товара соответствующего типа ($0 \leq a_i \leq 100$, $0 \leq b_i \leq 99$). Стоимость товара всегда больше нуля.

Формат выходных данных

Если требуемое количество копеечных монет получить невозможно, выведите -1 . Иначе выведите минимальную сумму, которую должен потратить Петя на покупку товаров, чтобы получить N однокопеечных монет. Сумма должна быть выведена как два целых числа, задающих рубли и копейки (второе число обязано быть от 0 до 99).

Примеры

f.in	f.out
3 1 0 2	0 2
4 2 1 2 0 4	0 16
1 3 0 1 0 4 0 6	0 1

Частичные ограничения

Первая группа состоит из тестов, в которых $N \leq 15$, $M \leq 10$.

Вторая группа состоит из тестов, в которых $N \leq 100$, $M \leq 100$.

Задача G. На санях

Имя входного файла: g.in
 Имя выходного файла: g.out
 Максимальное время работы на одном тесте: 2 секунды
 Максимальный объем используемой памяти: 64 мегабайта

В начале XIX века еще не было самолетов, поездов и автомобилей, поэтому все междугородние зимние поездки совершались на санях. Как известно, с дорогами в России тогда было даже больше проблем, чем сейчас, а именно на N существовавших тогда городов имелась ровно $N-1$ дорога, каждая из которых соединяла ровно два города. К счастью, из каждого города можно было добраться в любой другой (возможно, через некоторые промежуточные города). В каждом городе имелась почтовая станция (или, как ее называют, «ям»), на которой можно было пересест в другие сани. При этом ямщики могли долго запрягать (для каждого из городов известно время, которое ямщики в этом городе тратят на подготовку саней к поездке) и быстро ехать (также для каждого города известна скорость, с которой ездят ямщики из него). Можно считать, что количество ямщиков в каждом городе не ограничено.

Если бы олимпиада проводилась 200 лет назад, то путь участников занимал бы гораздо большее время, чем сейчас. Допустим, из каждого города в Москву выезжает участник олимпиады и хочет добраться до Москвы за наименьшее время (не обязательно по кратчайшему пути: он может заезжать в любые города, через один и тот же город можно проезжать несколько раз). Сначала он едет на ямщике своего города. Приехав в любой город, он может либо сразу ехать дальше, либо пересесть. В первом случае он едет с той же скоростью, с какой ехал раньше. Решив сменить ямщика, он сначала ждет, пока ямщик подготовит сани, и только потом едет с ним (естественно, с той скоростью, с которой ездит этот ямщик). В пути можно делать сколько угодно пересадок.

Жюри стало интересно, какое время необходимо, чтобы все участники олимпиады доехали из своего города в Москву 200 лет назад. Все участники выезжают из своих городов одновременно.

Формат входных данных

В первой строке входного файла дано натуральное число N , не превышающее 2000 — количество городов, соединенных дорогами. Город с номером 1 является столицей.

Следующие N строк содержат по два целых числа: T_i и V_i — время подготовки саней в городе i , выраженное в часах, и скорость, с которой ездят ямщики из города i , в километрах в час ($0 \leq T_i \leq 100$, $1 \leq V_i \leq 100$).

Следующие $N-1$ строк содержат описания дорог того времени. Каждое описание состоит из трех чисел A_j , B_j и S_j , где A_j и B_j — номера соединенных городов, а S_j — расстояние между ними в километрах ($1 \leq A_j \leq N$, $1 \leq B_j \leq N$, $A_j \neq B_j$, $1 \leq S_j \leq 10000$). Все дороги двусторонние, то есть если из A можно проехать в B , то из B можно проехать в A . Гарантируется, что из всех городов можно добраться в столицу.

Формат выходных данных

Сначала выведите одно вещественное число — время в часах, в которое в Москву приедет последний участник.

Далее выведите путь участника, который приедет самым последним (если таких участников несколько, выведите путь любого из них). Выведите город, из которого этот участник выехал первоначально, и перечислите в порядке посещения те города, в которых он делал пересадки. Последовательность должна заканчиваться столицей.

При проверке ответ будет засчитан, если из трех величин «время путешествия по выведенному пути», «выведенное время» и «правильный ответ» каждые две отличаются менее чем на 0.0001 .

Пример

g.in	g.out	Комментарий
4 1 1 10 30 5 40 1 10 1 2 300 1 3 400 2 4 100	31.0000 4 2 1	Участник из города 1 уже находится на своем месте и тратит на дорогу 0 часов. Участник из города 2 ждет 10 часов ямщика в своем городе, а затем проезжает 300 км от города 2 до города 1 за 10 часов, т.е. тратит на дорогу 20 часов. Участник из города номер 3 ждет ямщика 5 часов, а затем доезжает до города 1 за 10 часов, т.е. тратит на дорогу 15 часов. Участник из города 4 может доехать до города 1 с ямщиком из города 4 за $1 + 40 = 41$ час или доехать до города номер 2 за $1 + 10 = 11$ часов, прождать там 10 и доехать до столицы за 10 часов. Таким образом, он может добраться до города 1 минимум за 31 час. Это и есть самое большое время и ответ к задаче.
3 1 1 0 10 0 55 1 2 100 2 3 10	3.0000 2 3 1	Участнику из города 2 выгоднее добраться сначала до третьего города, где ездят быстрее, а потом поехать в столицу, не делая пересадки в своём городе.

Частичные ограничения

Первая группа состоит из тестов, в которых $N \leq 40$.

Вторая группа состоит из тестов, в которых $N \leq 400$.

Задача Н. Кодировка

Имя входного файла: h.in
Имя выходного файла: h.out
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 256 мегабайт

Вот уже три года майор Пронин ведёт борьбу с подпольной группой по производству контрафактной продукции. Недавно его коллега, внедрившийся под прикрытием в эту организацию, передал ему важнейший документ, описывающий местонахождение главной фабрики преступников. К несчастью, сразу после этого он был раскрыт.

Майор понимал, что у него совсем немного времени до того, как придут и за ним. Поэтому он выбрал некоторый кусок документа и переслал Вам, своему ближайшему соратнику по борьбе со злом (майор боялся, что не хватит времени отправить документ целиком). Сразу после этого к нему ворвались злоумышленники и схватили Пронина. Чтобы не вызывать никаких подозрений, преступники решили не удалять документ, а просто закодировать его некоторым образом. Известно лишь, что каждый символ был заменён ровно на один символ, при этом разные символы перешли в разные, одинаковые – в одинаковые. Известно, что и до, и после перекодировки документ состоял только из символов с ASCII-кодами от 32 до 255 включительно.

Получив часть документа S , Вы немедленно отправились выручать Пронина из беды, но его дом оказался пуст. На его компьютере остался закодированный документ. Ни майор, ни его коллега на связь не выходят. Вы поняли, что единственный способ отыскать их – проникнуть на главную фабрику организации. Для этого необходимо раскодировать весь документ. Прежде чем передать его лингвистам, нужно произвести техническую обработку.

Возможно, что в тексте есть несколько фрагментов, которые могут являться кодом имеющегося у вас фрагмента S . Вы наудачу предположили, что имеющийся у вас фрагмент S в закодированном тексте является самым первым из фрагментов, в которые он мог быть закодирован.

Ваша задача – вычислить, какие символы документа могут быть определены однозначно исходя из приведённых соображений.

Формат входных данных

В первой строке записан перекодированный документ, во второй – часть документа S , оказавшаяся у вас. Обе строки имеют длину не более 10^6 символов и состоят только из символов с ASCII-кодами от 32 до 255 включительно. Длина S меньше длины документа.

Формат выходных данных

Если на каком-то этапе произошла ошибка и входные данные некорректны, выведите Impossible. Иначе запишите в первой строку Possible, а во вторую – результат раскодировки. Символы, которые невозможно определить однозначно, замените на ?.

Примеры

h.in	h.out
ababc ab	Possible abab?
fadacabba bcc	Possible ?b?b?bccb
abcdef ee	Impossible
cdcd ab	Possible abab

Частичные ограничения

Первая группа состоит из тестов, в которых текст и образец имеют длину не более 100 и могут состоять лишь из букв a, b, c, d, e, f.

Вторая группа состоит из тестов, в которых текст и образец имеют длину не более 1000.