

Задача А. Завоеватель

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

Первый всадник Апокалипсиса Завоеватель пришел на Землю. Увидели это жители одного города и решили предупредить свою столицу, дабы ее жители успели подготовиться. С этой целью жителями города в столицу был отправлен гонец на лошади.

Дорога между этим городом и столицей представляет собой прямую, на которой, включая этот город и столицу, расположены n городов, причем расстояние между любыми соседними городами на этой прямой одинаково. В каждом из $n - 2$ городов, мимо которых гонцу необходимо проехать, он может сменить лошадь в конюшне. Хозяин каждой конюшни знает, сколько минут требуется лошади в этой конюшне на дорогу между двумя соседними городами. За какое минимальное время гонец сможет добраться до столицы (n -го города на этой прямой)?

Формат входного файла

В первой строке входного файла содержится одно целое число n ($1 \leq n \leq 100\,000$) — количество городов. В следующей строке содержатся n натуральных чисел t_i ($1 \leq t_i \leq 10^6$) — количество минут, необходимое лошади из конюшни в i -м городе на преодоление расстояния между двумя городами. Заметим, что первое число означает скорость лошади, которая была у гонца при выезде из первого города, а последнее — скорость лошади в конюшне столицы.

Формат выходного файла

В выходной файл выведите одно число — количество минут, за которое гонец доедет до столицы.

Примеры

<code>conquest.in</code>	<code>conquest.out</code>
6 3 4 3 2 1 5	12

Задача В. k-сортировка

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

В этом году Гриша поступил в Университет ИТ. В Университете ИТ очень много новых предметов, интересных и не очень. Особенно Грише нравится предмет «Алгоритмы и структуры данных». На последней лекции были рассказаны алгоритмы сортировки. Гриша — очень амбициозный молодой человек и хочет изобрести свой алгоритм, который впоследствии будет назван именем его любимого дедушки. Вдохновившись чтением многотомника Кнута, Гриша решил модернизировать какой-нибудь уже существующий алгоритм сортировки натуральных чисел, наложив следующее ограничение. Любые два элемента можно менять местами, только если они сравнимы по модулю некоторого натурального числа k , то есть дают одинаковые остатки при делении на k . Но все инновационные методы требуют проверки, поэтому Гриша обратился за помощью к Вам!

Проверьте, сможет ли новая версия алгоритма отсортировать заданный массив натуральных чисел.

Формат входного файла

Первая строка входного файла содержит два числа n ($1 \leq n \leq 1000$) и k ($1 \leq k \leq 10^9$) — количество элементов в массиве и число, по модулю которого сравниваются элементы массива.

Вторая строка входного файла содержит n целых чисел a_i — элементы массива ($1 \leq a_i \leq 10^9$).

Формат выходного файла

В выходной файл выведите «YES», если алгоритм сможет отсортировать заданный массив и «NO» — в обратном случае.

Примеры

<code>ksort.in</code>	<code>ksort.out</code>
5 2 5 4 3 2 1	YES
3 2 2 3 1	NO

Задача С. Марсианский друг

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

Десятиклассник Вася дружит с марсианином Хрулем. И сегодня Хруль попросил Васю помочь решить следующая задачу: необходимо найти на какое количество нулей заканчивается число $N!$. Вася сначала обрадовался, ведь он решал такую задачу в пятом классе на олимпиадном кружке по математике. Но вовремя вспомнил, что на Марсе k -ичная система счисления, и задумался.

Помогите Васе посчитать количество нулей в записи число $N!$ в k -ичной системе счисления.

Формат входного файла

Входной файл содержит два целых числа N ($1 \leq N \leq 10^{18}$) и k ($2 \leq k \leq 10^9$).

Формат выходного файла

В выходной файл выведите одно целое число — ответ на задачу.

Примеры

<code>mars.in</code>	<code>mars.out</code>
20 7	2

Задача D. Кодовый замок

Имя входного файла: `lock.in`
Имя выходного файла: `lock.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Чтобы проникнуть на секретную базу, на которой скрывается Доктор Но, Джеймсу Бонду необходимо взломать кодовый замок. Поскольку знание криптографии и взлом замков не входит в должностные обязанности агента 007, он обратился к вам за помощью.

У кодового замка n табло, на каждом из которых написано некоторое число a_i . Кроме этого, под каждым табло, кроме последнего, есть большая красная кнопка. Исследования Джеймса Бонда показали, что при нажатии кнопки, расположенной под табло номер i , вместо числа, которое было написано на этом табло, на нем появляется число, написанное в этот момент на табло номер $i + 1$.

С помощью своего недюжинного обаяния Бонду удалось выяснить, что попасть на базу у него получится только тогда, когда последовательность чисел, написанных на табло, станет неубывающей. Теперь он хочет выяснить, за какое минимальное количество нажатий на кнопки он сможет добиться такой ситуации.

Формат входного файла

В первой строке входного файла дано одно целое число n ($1 \leq n \leq 100\,000$) — количество табло с числами. В следующей строке перечислены n целых чисел a_i ($1 \leq a_i \leq 100\,000$) — числа, написанные на табло до начала взлома.

Формат выходного файла

В первой строке выходного файла выведите одно целое число — ответ на задачу.

Примеры

<code>lock.in</code>	<code>lock.out</code>
6 1 4 2 2 5 3	2

Задача Е. Поручения

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

Учитель Сплинтер всегда держит своих учеников в тонусе. Он дал им n поручений: им нужно помочь Кейси Джонсу в уличной драке, спасти Землю от нападок Шреддера и сорвать коварные планы Кренга, а в довесок еще сделать кучу дел по дому. Причем черепашкам необходимо выполнить все эти задания.

Очевидно, каждое поручение — не из приятных и доставляет черепашкам какое-то количество боли и страданий. Когда черепашки выполняют очередное задание, боль, которую оно приносит, может добавиться к усталости черепашек. Однако, это происходит только в том случае, если любое из заданий, выполненных ими раньше, приносило им меньше боли, чем последнее выполненное. Страдание добавляется к усталости по таким же правилам.

Теперь черепахи хотят выполнять задания в таком порядке, чтобы после выполнения всех заданий усталость была минимальна.

Формат входного файла

В первой строке входного файла задано одно целое число n — количество заданий, которые получили черепашки. ($1 \leq n \leq 10^5$). Далее следуют n строк, где для каждого i -го задания задано два целых числа — количество боли a_i и страдания b_i ($1 \leq a_i, b_i \leq 10^9$). Гарантируется, что все a_i различны и все b_i различны.

Формат выходного файла

В первую строку выходного файла выведите минимальную усталость черепашек после выполнения всех заданий. Во второй строчке выходного файла выведите перестановку чисел от 1 до n — порядок, в котором следует выполнять задания. Если существует несколько оптимальных ответов, выведите любой.

Примеры

<code>missions.in</code>	<code>missions.out</code>
3	8
3 2	2 1 3
2 3	
1 1	

Задача F. Химический шифр

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

Борис очень любит химию. Причина проста: знание этой науки позволяет в домашних условиях синтезировать невидимые чернила, яды и взрывчатку — в общем все то, что может быть интересно здоровому подростку.

Борис не хочет, чтобы кто-либо кроме него мог читать записи его экспериментов, потому решил применять специальный шифр. Для того, чтобы записать несколько слов, он выписывает их подряд, а затем возможно вычеркивает несколько букв из получившейся строки.

Недавно Борису понадобилось воспроизвести один из своих экспериментов по синтезу кристалла с квадратной решеткой. После изучения дневника он с ужасом осознал, что не может по шифру восстановить набор химических элементов, присутствующих в кристалле. Борис точно помнит, что их было не очень много, поэтому просит вас определить минимальное число элементов, которые могут давать шифр, записанный в дневнике.

Борис готов предоставить список всех когда либо использованных им химических элементов. Кроме того известно, что по рассеянности Борис мог записать один и тот же элемент несколько раз. В этом случае нужно считать каждое его вхождение.

Формат входного файла

Первая строка входного файла содержит шифр — строку s ($1 \leq |s| \leq 100$). Вторая строка входного файла содержит одно целое число n — количество химических элементов, которые мог использовать Борис ($1 \leq n \leq 100$). Следующие n строк содержат по одному обозначению элемента, состоящему из одного или двух символов.

Все строки во входном файле состоят из строчных и прописных букв латинского алфавита. Прописные и строчные буквы считаются различными.

Формат выходного файла

Выведите одно число — минимальное количество химических элементов, которые могут давать необходимый шифр. Если ответа не существует, выведите «-1».

Примеры

<code>chemcode.in</code>	<code>chemcode.out</code>
FeCuAg 2 0 H	-1
FeCgCu 3 Fe Cu Ag	4

Задача G. СМС

Имя входного файла:	<code>sms.in</code>
Имя выходного файла:	<code>sms.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Одним из самых популярных способов использования мобильной связи являются СМС-сообщения. Каждый день миллионы людей отправляют десятки миллионов сообщений: «Привет!», «Как дела?», «Ты где?», «Я задержусь» и еще тысячи различных коротких посланий.

К сожалению, на клавиатуре мобильного телефона может оказаться меньше кнопок, чем букв в алфавите. Поэтому на первой кнопке размещаются несколько первых букв алфавита, на второй — следующие несколько и так далее. Чтобы набрать некую букву, необходимо нажать ту кнопку, на которой она размещена, t раз, если буква является t -ой по алфавиту, размещенной на этой кнопке. Так, если на первой кнопке клавиатуры размещены буквы a , b и c , то для выбора буквы c придется нажать эту кнопку трижды.

Ученые изучили среднее количество раз, которое каждая буква алфавита встречается в СМС-ках за время службы среднестатистического телефона. Теперь у фирмы-производителя появилась возможность спроектировать клавиатуру так, чтобы минимизировать суммарное количество нажатий на все кнопки. Помогите им сделать это.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \leq n \leq 30$, $1 \leq m \leq 10$) — количество букв в алфавите и кнопок на клавиатуре телефона.

Следующая строка содержит n целых чисел a_i ($0 \leq a_i \leq 1\,000\,000$) — количество раз, которые будет напечатана i -ая буква алфавита.

Формат выходного файла

В выходной файл выведите m целых неотрицательных чисел b_i — количество букв, отнесенных к i -ой кнопке клавиатуры. Сумма всех b_i должна быть равна n .

Если возможных ответов несколько — выведите любой.

Примеры

<code>sms.in</code>	<code>sms.out</code>
5 3 1 3 5 1 1	1 1 3

Note

В примере на первой кнопке размещена одна первая буква алфавита, на второй — только вторая, а на третьей — оставшиеся три.

Таким образом, суммарное количество нажатий на кнопки будет: $1 \times 1 + 3 \times 1 + 5 \times 1 + 1 \times 2 + 1 \times 3 = 14$.

Задача Н. Канализация

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

Как известно, черепашки-ниндзя вместе со своим учителем Сплинтером всю свою жизнь живут в канализации. Где же еще можно так быстро передвигаться по городу, скользая на панцире?

Из канализации черепашек в город можно выбрать только с помощью n канализационных люков, расположенных в разных частях города. Некоторые люки соединены друг с другом трубами, по которым черепашки могут передвигаться в обе стороны. Между каждой парой люков существует ровно один путь по трубам. Это, в частности, значит, что в канализации ровно $n - 1$ труба. Люки пронумерованы начиная с 1.

Когда черепашки-ниндзя были маленькими, они постоянно забывали маршруты от одного люка до другого, и спрашивали учителя Сплинтера, куда же им скользить, чтобы попасть в какой-то люк. Сплинтер сообщал черепашкам номер следующего люка на пути из люка l к люку номер r .

Вскоре учителю Сплинтеру надоело, что черепашки постоянно отвлекают его от медитации, и они вместе с Донателло написали программу, которая по двум номерам люков l и r сообщает номер следующего люка на пути из l в r . А вы сможете написать такую программу?

Формат входного файла

В первой строке входного файла заданы два целых числа n и m — количество люков в канализации и количество запросов соответственно ($1 \leq n, m \leq 10^5$).

В следующих $n - 1$ строке заданы описания труб — два числа a и b , которые задают номера люков, которые соединяет эта труба ($1 \leq a, b \leq n$). По трубе можно скользить в обоих направлениях. Гарантируется, что в канализации можно добраться от каждого люка до каждого.

В следующих m строках заданы запросы, по одному запросу в строке — два числа l и r , которые задают стартовый и конечный люк на пути соответственно ($1 \leq l, r \leq n$). Гарантируется, что l и r — различны.

Формат выходного файла

В выходной файл выведите ответы на запросы по одному в строке, в том порядке, в котором они следуют во входном файле. Ответом на запрос является номер следующего люка на пути с l по r .

Примеры

<code>canalization.in</code>	<code>canalization.out</code>
6 4	2
1 2	2
3 2	6
4 2	5
2 5	
5 6	
1 5	
4 6	
5 6	
6 3	

Задача I. Прыгать!

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

Тигра любит прыгать! К сожалению, прыгать по лесу не очень удобно — и за кочку можно зацепиться и в дерево врезаться. Умный и добрый Кролик решил помочь Тигре и построить несколько удобных дорог из желтого кирпича (ведь желтый — любимый цвет Тигры). Для этого он составил полную карту Стоакрового Леса и посчитал какие дороги можно построить, и сколько кирпича на каждую из них потребуется. Оценив запасы кирпича, Кролик понял что возможно вместо некоторых дорог можно построить настоящие лесные автобаны — широкие дороги, где Тигра сможет разогнаться еще быстрее и получить больше удовольствия от движения. Для постройки автобана вместо обычной дороги потребуется в c раз больше кирпича, независимо от расположения дороги.

Каждая дорога соединяет какие-то два интересных для Тигры места — домики обитателей Леса, поляны, где он может вдоволь попрыгать и так далее. Кролик пронумеровал интересные места целыми числами от 1 до n и выписал список возможных дорог. Теперь он хочет построить дорожную сеть таким образом, чтобы было построено как можно больше автобанов, но при этом можно было из каждого интересного места попасть в любое другое. Для реализации этого плана у Кролика в распоряжении есть k кирпичей. Помогите ему спланировать постройку дорог!

Формат входного файла

Первая строка входного файла содержит четыре целых числа n , m , k и c ($1 \leq n, m \leq 100000$; $1 \leq k \leq 10^9$; $1 \leq c \leq 1000$) — количество интересных мест, возможных дорог, общее количество кирпичей и коэффициент, во сколько раз больше кирпичей требуется на постройку автобана, соответственно.

Следующие m строк содержат описания дорог — три целых числа a_i , b_i , l_i ($1 \leq a_i, b_i \leq n$; $1 \leq l_i \leq 10^6$) — номера интересных мест, которые соединяет дорога и количество кирпича, необходимое на постройку этой дороги, соответственно.

Каждая дорога соединяет два различных интересных места. Между двумя интересными местами может быть более одной дороги.

Формат выходного файла

Если невозможно построить такую сеть дорог, то выведите в выходной файл единственное слово «Impossible».

Иначе, в первой строке выходного файла выведите два целых числа p и q — количество обычных дорог и автобанов в оптимальном плане. Во второй строке выведите p целых чисел — номера обычных дорог, которые необходимо построить в возрастающем порядке. В третьей строке выведите q целых чисел — номера автобанов, также в возрастающем порядке.

Дороги пронумерованы в порядке появления во входном файле. В случае нескольких оптимальных решений, выведите любое.

Примеры

jump.in	jump.out
4 2 10 2 1 2 3 3 4 5	Impossible
4 4 10 2 1 2 3 3 4 5 1 3 1 3 2 1	1 2 2 3 4

Задача J. Конфетки

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

У Кролика день рождения! Он пригласил в гости n гостей. Чтобы гостям не было грустно и скучно, Кролик купил n коробок конфет. Кролик любит разнообразие, поэтому конфеты были разные. В i -й коробке лежало a_i конфет.

В назначенный день с самого утра к Кролику начали приходить гости. Каждый гость характеризуется своей наглостью b_i . Это означает, что, зайдя домой к Кролику и увидев коробки конфет, он брал из каждой коробки, в которой не меньше, чем b_i , конфет, по одной и съедал её. Например, у Винни-Пуха вполне могла быть наглость один. Это значит, что он бы съел по конфете из каждой коробки.

Вечером, когда гости разошлись, Кролику стало интересно, кто съел сколько конфет. Помогите ему определить это.

Формат входного файла

В первой строке задано целое число n ($1 \leq n \leq 100\,000$) — количество коробок конфет. В следующей строке задано n натуральных чисел a_i ($1 \leq a_i \leq 10^9$) — сколько конфет в каждой коробке.

Далее, в следующей строке задано число m ($1 \leq m \leq 100\,000$) — количество гостей. В четвёртой и последней строке задано m чисел b_i ($1 \leq b_i \leq 10^9$) — наглости гостей.

Формат выходного файла

В выходной файл выведите n строк, i -ая из которых должна содержать количество конфет съеденных i -ым гостем.

Примеры

<code>candies.in</code>	<code>candies.out</code>
3	3
3 1 1	1
2	
1 1	

Задача К. Миньоны развлекаются

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

На карте лаборатории Грю отмечено n контрольных точек. Между ними проведено m двусторонних дорожек, на каждой из которых лежит несколько бананов. Миньонам предлагается начать свой путь в любой контрольной точке, побегать по дорожкам, причем по каждой из них можно пробежать не более одного раза, а потом вернуться в исходную точку. Пусть миньон пробежал какой-то такой циклический путь и прошел по ребрам, на которых лежало c_1, c_2, \dots, c_k бананов соответственно. Тогда он получит за этот путь количество очков, равное $\min(c_1, c_2, \dots, c_k) + \max(c_1, c_2, \dots, c_k)$.

Миньон Дэйв — один из участников этого соревнования, и он очень хочет победить. Поэтому он обратился за помощью, чтобы вы помогли ему найти оптимальный циклический путь, то есть путь, за который он получит максимальное количество очков. Не отказывайте этому милому созданию, помогите ему!

Формат входного файла

В первой строке входного файла даны два числа n, m ($1 \leq n, m \leq 100\,000$) — количество контрольных точек и количество дорожек между ними соответственно.

В следующих m строках дано описание дорожек между контрольными точками — в i -й из них написано три числа v, u, w ($1 \leq v, u \leq n; v \neq u; 0 \leq w \leq 10^9$) — номера контрольных точек, между которыми проходит i -я дорожка, и количество бананов на ней. Между одной парой контрольных точек может проходить несколько дорожек.

Формат выходного файла

В единственной строке выходного файла выведите ответ на задачу — максимальное значение суммы минимального количества бананов и максимального среди всех циклических путей.

Если циклического пути вообще нет, в единственной строке выходного файла выведите 0.

Пример

<code>cycle.in</code>	<code>cycle.out</code>
3 3 1 2 1 2 3 1 3 1 1	2
4 4 1 2 1 2 3 2 3 1 1 1 4 100	3
4 5 1 2 2 2 3 1 3 1 1 1 4 2 4 2 2	4
2 1 1 2 1	0