Задача A. Диофантово уравнение

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Даны натуральные числа a, b и c. Решите в целых числах уравнение $ax+by=c$. Среди множества решений следует выбрать такое, где x имеет наименьшее неотрицательное значение.

Формат входных данных

Первая строка содержит три целых числа a и b и c ($1 \leqslant a, b, c \leqslant 10^9$).

Формат выходных данных

Выведите искомые x и y через пробел. Если решения не существует, выведите одну строку «`Impossible`».

Примеры

стандартный ввод	стандартный вывод
1 2 3	1 1
10 6 8	2 -2

Задача B. Представление чисел

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Дано натуральное число N. Требуется представить его в виде суммы двух натуральных чисел A и B таких, что НОД (наибольший общий делитель) чисел A и B — максимален.

Формат входных данных

Во входном файле записано натуральное число N ($2 \leqslant N \leqslant 10^9$)

Формат выходных данных

В выходной файл выведите два искомых числа A и B. Если решений несколько, выведите любое из них.

Пример

стандартный ввод	стандартный вывод
100	50 50

Задача C. Кинотеатр

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

Марья Ивановна с Марьей Михайловной привели школьников в кинотеатр. Чтобы не было никаких обид, Марья Ивановна построила всех школьников по алфавиту и рассадила их: сначала в первый ряд слева направо, затем во второй слева направо и т.д., заполнив весь зал из n рядов по m кресел. Тут пришла Марья Михайловна и сказала, что ребята сели неправильно – надо пересесть. Она предложила сначала заполнить все первые места от первого ряда к последнему, затем все вторые места и т. д.

Определите, сколько школьников после такой пересадки останется на своем месте.

Например, если n = 3 и m = 3, то в первом случае дети сядут так:

1 2 3
4 5 6
7 8 9
а во втором – так:
1 4 7
2 5 8
3 6 9
Таким образом, три школьника: 1, 5 и 9 останутся на своих местах.

Формат входных данных

Вводятся два целых числа n и m ($1 \leqslant n, m \leqslant 10^9$).

Формат выходных данных

Выведите количество школьников, которые останутся на своих местах.

Примеры

стандартный ввод	стандартный вывод
`3 3`	`3`
`2 4`	`2`

Задача D. Наибольший общий делитель

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

Сегодня на уроке математики шестиклассник Петя изучил понятие наибольшего общего делителя. Петя тут же решил применить полученные знания на практике.

Петя выписал на листке бумаги n чисел $a_1, a_2, \ldots, a_n$ — номера домов, в которых живут его друзья. Теперь он хочет выбрать такое подмножество этих чисел, чтобы их наибольший общий делитель был равен его любимому числу d.

Помогите Пете выбрать из выписанных чисел искомое подмножество.

Формат входных данных

Первая строка входного файла содержит два целых числа n и d ($1 \leqslant n \leqslant 1000$, $1 \leqslant d \leqslant 10^9$). Вторая строка содержит n целых чисел: $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Если существует искомое подмножество, выведите на первой строке выходного файла число k — количество чисел в нем. На второй строке выведите числа, входящие в это подмножество.

Если решения не существует, выведите на первой строке выходного файла число -1.

Если возможных ответов несколько, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
4 3 6 8 12 9	2 6 9
3 3 2 4 8	-1

Задача E. Армия математиков

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

У вас есть n математиков. Пусть интеллектуальность i-го математика равна a_i. Для некоторого k назовём $i_1, i_2, \ldots, i_k$ сходкой математиков, если $i_1 < i_2 < i_3 < \ldots < i_k$ и $gcd(a_{i_1}, a_{i_2}, \ldots, a_{i_k}) > 1$. Эффективность этой сходки равна $k \cdot gcd(a_{i_1}, a_{i_2}, \ldots, a_{i_k})$.

Найдите сумму эффективностей всех сходок математиков. Так как это число может быть очень большим, выведите его по модулю 1000000007 ($10^9 + 7$).

Формат входных данных

Первая строка содержит целое число n ($1 \leqslant n \leqslant 200000$) — количество математиков.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 1000000$) — интеллектуальности математиков.

Формат выходных данных

Выведите одно число — ответ.

Примеры

стандартный ввод	стандартный вывод
3 3 3 1	12
4 2 3 4 6	39

Замечание

В первом примере сходки — 1, 2, 1, 2, так что ответ $1 \cdot 3 + 1 \cdot 3 + 2 \cdot 3 = 12$

Задача F. Чиселки и странные функции

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Есть функция $f(n)$, где $f(1) = 1$, а для $n \geqslant 2$, $f(n)$ равно количеству различных упорядоченных пар положительных целых чисел $(x,\ y)$ таких, что $x + y = n$ и $gcd(x, y) = 1$. Число $gcd(a, b)$ равно наибольшему общему делителю a и b.

Есть функция $g(n) = \sum_{d|n} f(n/d)$. Суммирование проводится по всем положительным целым числам d, делящим n.

Определим $F_k(n)$ так:

$$F_k(n) = \begin{cases} f(g(n)) & \text{для } k = 1 \\ g(F_{k-1}(n)) & \text{для } k > 1 \text{ и } k \bmod 2 = 0 \\ f(F_{k-1}(n)) & \text{для } k > 1 \text{ и } k \bmod 2 = 1 \end{cases}$$

Найдите $F_k(n)$ по модулю 100000007.

Формат входных данных

В единственной строке находятся два целых числа n ($1 \leqslant n \leqslant 10^{12}$) и k ($1 \leqslant k \leqslant 10^{12}$).

Формат выходных данных

Выведите одно целое число — значение $F_k(n)$ по модулю 100000007.

Примеры

стандартный ввод	стандартный вывод
`7 1`	`6`
`10 2`	`4`

Замечание

В первом примере есть 6 различных упорядоченных пар $(1, 6)$, $(2, 5)$, $(3, 4)$, $(4, 3)$, $(5, 2)$ и $(6, 1)$, удовлетворяющих $x + y = 7$ и $gcd(x, y) = 1$. Поэтому $f(7) = 6$. В итоге, $F_1(7) = f(g(7)) = f(f(7) + f(1)) = f(6 + 1) = f(7) = 6$.

Задача G. Функция Эйлера

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

Красить забор — не очень. Вернёмся к математике.

Формат входных данных

Дано число n ($1 \leqslant n \leqslant 10^8$).

Формат выходных данных

Для каждого числа от 1 до n требуется посчитать функцию Эйлера от него. Так как чисел очень много, сначала выведите сумму функций Эйлера для первых 100 чисел, потом для вторых 100 чисел, потом для третьих 100 чисел и так далее. Если n не делится на 100, последнее из выведенных вами чисел будет состоять из суммы меньше, чем 100 слагаемых.

Примеры

стандартный ввод	стандартный вывод
10	32
200	3044 9188

Замечание

Для чисел от 1 до 10 функция Эйлера будет равна соответственно $1, 1, 2, 2, 4, 2, 6, 4, 6, 4$, что в сумме даёт 32.

Задача H. Функция Эйлера [Мало времени и памяти!]

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2.1 секунд
Ограничение по памяти:	512 мегабайт

Как насчет позагонять?

Формат входных данных

Дано число n ($1 \leqslant n \leqslant 10^8$).

Формат выходных данных

Для каждого числа от 1 до n требуется посчитать функцию Эйлера от него. Так как чисел очень много, сначала выведите сумму функций Эйлера для первых 100 чисел, потом для вторых 100 чисел, потом для третьих 100 чисел и так далее. Если n не делится на 100, последнее из выведенных вами чисел будет состоять из суммы меньше, чем 100 слагаемых.

Примеры

стандартный ввод	стандартный вывод
`10`	`32`
`200`	`3044 9188`

Замечание

Для чисел от 1 до 10 функция Эйлера будет равна соответственно $1, 1, 2, 2, 4, 2, 6, 4, 6, 4$, что в сумме даёт 32.

Задача I. Сколько простых?

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

Найдите количество простых чисел от n^2 до $n^2 + n$ включительно.

Формат входных данных

Первая строка содержит число n ($1 \leqslant n \leqslant 10^7$).

Формат выходных данных

Выведите количество простых чисел от n^2 до $n^2 + n$ включительно.

Примеры

стандартный ввод	стандартный вывод
2	1
5	1

Задача J. Доставка посылок

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3.5 секунд
Ограничение по памяти:	512 мегабайт

В городе Nске есть n домов, которые соединены $n - 1$ двусторонней дорогой так, что от любого дома можно доехать до любого другого, передвигаясь по дорогам. В каждом доме проживает ровно один человек.

В один праздничный день в качестве подарка каждый житель города решил отправить каждому посылку. Посылки в Nске доставляет специальная бесконтактная почта. Чтобы доставить жителю дома u посылку от жителя дома v, почтовый курьер забирает посылку около дома v, затем едет по кратчайшему пути до дома u и оставляет посылку около него.

К сожалению, в Nске действуют строгие правила отправки посылок. Вес каждой посылки должен быть целочисленным, а также для каждой дороги установлено целое число w_i — ограничение на провоз посылок, означающее, что по этой дороге можно провозить только посылки, вес которых делит w_i. Иными словами, посылка веса x может быть провезена по дороге с ограничением w_i только если w_i делится нацело на x.

Жители города очень любят праздники, поэтому каждый житель решил отправить каждому посылку наибольшего возможного веса.

Чтобы обеспечить наилучшую работу бесконтактной почты, необходимо посчитать суммарный вес всех отправленных посылок. Помогите решить эту важную задачу!

Формат входных данных

Первая строка содержит целое число n ($2 \leqslant n \leqslant 100\,000$) — количество домов в городе.

Каждая из следующих $n - 1$ строк содержит описание дороги: три целых числа u_i, v_i и w_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$, $1 \leqslant w_i \leqslant 10^6$) — номера домов, между которыми проложена i-я дорога, и ограничение на провоз посылок в i-й дороге. Дома в городе нумеруются, начиная с единицы.

Гарантируется, что между каждой парой домов существует путь по дорогам.

Формат выходных данных

Выведите единственное число — сумму весов всех отправленных жителями посылок.

Примеры

стандартный ввод	стандартный вывод
3 1 2 6 1 3 4	24
6 1 2 4 3 4 3 1 4 10 5 1 2 4 6 5	82

Замечание

В первом примере из условия житель 1 мог отправить жителю 2 посылку веса не более 6, аналогично житель 2 мог отправить жителю 1 посылку веса не более 6. Житель 1 мог отправить жителю 3 посылку веса не более 4, аналогично житель 3 мог отправить жителю 1 посылку веса не более 4. Житель 2 мог отправить жителю 3 посылку веса не более 2, так как число 6 должно делиться на вес посылки, и число 4 должно делиться на вес посылки. Аналогично житель 3 мог отправить жителю 2 посылку веса не более 2.

В таблице ниже отмечены максимальные веса, которые могут отправить друг другу жители во втором примере из условия.

Житель	1	2	3	4	5	6
1	–	4	1	10	2	5
2	4	–	1	2	2	1
3	1	1	–	3	1	1
4	10	2	3	–	2	5
5	2	2	1	2	–	1
6	5	1	1	5	1	–

Система оценки

Тесты к этой задаче состоят из восьми групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

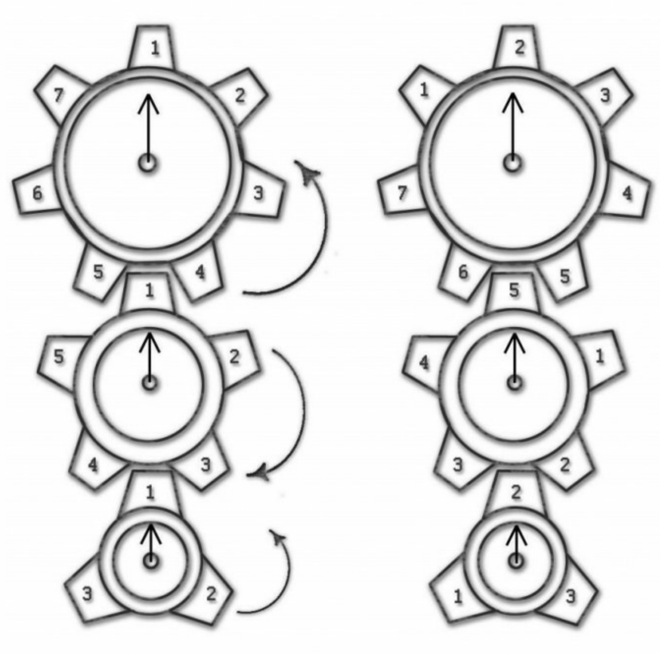
Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n	w_i		
0	0	–	–	–	Тесты из условия.
1	14	$n \leqslant 500$	$w_i \leqslant 100\,000$	0	
2	11	$n \leqslant 3000$	–	0, 1	
3	12	–	–	–	Все w_i простые числа.*
4	15	–	–	–	$u_i = 1, v_i = i + 1$
5	16	–	–	–	$u_i = i, v_i = i + 1$
6	10	–	$w_i \leqslant 500$	0	
7	8	–	$w_i \leqslant 10\,000$	0, 6	**Offline-проверка.**
8	14	–	–	0–7	**Offline-проверка.**

*Простым числом называется число, имеющее ровно два различных натуральных делителя — единицу и самого себя.

Задача K. Шестеренки

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

На каждой из трех осей установлено по одной вращающейся шестеренке и неподвижной стрелке. Шестеренки соединены последовательно. На первой шестеренке n зубцов, на второй — m, на третьей — k. На каждом зубце первой, второй и третьей шестеренок по часовой стрелке написаны в порядке возрастания числа от 1 до n, от 1 до m и от 1 до k соответственно. Стрелки зафиксировали таким образом, что когда стрелка первой оси указывает на число, стрелки двух других осей также указывают на числа. Витя записывает три числа (a_1, a_2, a_3), на которые указывают стрелки. После этого он поворачивает первую шестеренку на угол $360°/n$ против часовой стрелки, чтобы напротив стрелки на первой оси оказался следующий (по часовой стрелке) зубец. При этом вторая шестеренка поворачивается на угол $360°/m$ по часовой стрелке (размеры зубцов у шестеренок одинаковые, поэтому размеры самих шестеренок разные, чтобы на границе шестеренок равномерно уложилось разное число одинаковых по размеру зубцов), а третья шестеренка поворачивается на угол $360°/k$ против часовой стрелки. Витя опять записывает три числа, на которые указывают стрелки.



Поступая и далее таким образом, Витя заметил, что после некоторого количества таких действий стрелки показывают на три первоначальных числа.

Теперь его очень интересует, как по двум данным тройкам чисел определить, принадлежат ли они к одной последовательности, то есть можно ли некоторым количеством поворотов перейти от первой тройки ко второй. Он попросил вас написать программу, которая отвечает на этот вопрос.

Формат входных данных

В первой строке содержатся четыре числа T, n, m, k ($1 \leqslant T \leqslant 10, 1 \leqslant n, m, k \leqslant 10^{18}$) — количество пар троек, которые хочет проверить Витя и количества зубцов соответственно на первой, второй и третьей шестеренках.

В следующих T строках записаны шесть натуральных чисел a_1, a_2, a_3, b_1, b_2, b_3 ($1 \leqslant a_1, b_1 \leqslant n$, $1 \leqslant a_2, b_2 \leqslant m$, $1 \leqslant a_3, b_3 \leqslant k$).

Формат выходных данных

Для каждой пары троек в выходной файл выведите «YES», если обе тройки принадлежат одной последовательности, и «NO» иначе. Каждое слово должно быть в отдельной строке, в порядке, соответствующем входному файлу.

Примеры

стандартный ввод	стандартный вывод
3 11 13 15 5 5 5 6 4 6 11 13 15 1 12 1 2 13 2 1 1 1	YES YES YES
2 2 2 2 1 1 1 1 1 2 1 1 1 2 2 2	NO YES
1 7 5 3 1 1 1 2 1 1	YES

Замечание

Пояснение к примерам.

В первом примере в первой и второй парах вторая тройка получается из первой за один поворот первой шестеренки против часовой стрелки. В третьем случае из второй конфигурации просто получить первую опять же одним поворотом первой шестеренки против часовой стрелки. Очевидно, что тогда из первой можно каким-то образом получить вторую.

Во втором примере в первой паре тройки нельзя перевести друг в друга. Во второй тройки переходят друг в друга при одном повороте.

В третьем примере (1, 1, 1) — (семь поворотов первой против часовой стрелки шестеренки) — (1, 4, 2) — (еще семь таких же поворотов) — (1, 2, 3) — (один поворот) — (2, 1, 1)

Задача L. k-суммы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	64 мегабайта

Неизвестный массив состоит из n целых чисел. k-сумма этого массива получается разделением его на подотрезки длины k и суммированием чисел в каждом из подотрезков. Если n не делится на k нацело, то последний подотрезок содержит меньше чем k слагаемых. Другими словами, k-сумма — массив, который можно представить как $(x[1] + \ldots + x[k])$, $(x[k+1] + \ldots + x[2k])$ и так далее, где последняя сумма, содержащая $x[n]$, может состоять из менее чем k слагаемых. Например, 5-сумма массива из 13 элементов состоит трех сумм (сумма элементов 1-5, сумма элементов 6-10 и сумма элементов 11-13). Очевидно, что нельзя однозначно восстановить изначальный массив по одной его k-сумме, но это становится возможным, если известны несколько его k-сумм для разных k.

Для заданного n и множества $k_1, k_2, \ldots, k_m$, определите сколько элементов изначального массива можно было бы восстановить, если бы были известны все слагаемые каждой k-суммы. Не составляет труда показать, что количество восстановленных элементов не зависит от самих слагаемых.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m — длина изначального массива и количество k-сумм ($1 \leqslant n \leqslant 10^9$, $1 \leqslant m \leqslant 10$).

Вторая строка содержит m различных целых чисел $k_1, k_2, \ldots, k_m$ ($1 \leqslant k_i \leqslant n$).

Формат выходных данных

Выведите единственное целое число — количество элементов изначального массива, которые можно было бы однозначно восстановить, если были бы известны все слагаемые каждой k-суммы.

Примеры

стандартный ввод	стандартный вывод
3 1 2	1
6 2 2 3	2
123456789 3 5 6 9	10973937

Задача M. Сколько простых 2?

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	64 мегабайта

Найдите количество простых чисел от 1 до n включительно.

Формат входных данных

Первая строка содержит число n ($1 \leqslant n \leqslant 10^8$).

Формат выходных данных

Выведите количество простых чисел от 1 до n включительно.

Примеры

стандартный ввод	стандартный вывод
2	1
5	3