

Задача А. Пробка

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

На односторонней дороге стоит светофор. Также, на этой дороге стоят n машин, которые хотят поскорее проехать. Однако, природная законопослушность мешает водителям ехать на красный и им приходится ждать зелёного, прежде чем ехать.

На светофоре есть только красный и зелёный цвета. Время, которым каждый свет горит, постоянно. Зелёный горит g секунд, затем красный горит r секунд, затем снова зелёный g секунд, и.т.д. В начальный момент светофор начинает светить зелёным. В момент переключения ехать можно.

На часах семь утра, все водители хотят спать, поэтому, они не сразу понимают, что уже можно ехать. А именно, для того, чтобы понять, что можно ехать, у i -го водителя уходит a_i секунд. То есть, он сможет поехать только через a_i секунд после того, как загорится зелёный, если он первый перед светофором, или после того как начнёт движение стоящий перед ним автомобиль, если в очереди он не первый. Автомобили ездят быстро, поэтому временем проезда светофора и временем подъезда к нему можно пренебречь. Несмотря на свою сонность, на красный свет водители реагируют мгновенно: ни один автомобиль не проедет на красный свет. Определите, через сколько секунд после первого включения зеленого свет уедет последний автомобиль.

Формат входного файла

В первой строке заданы целые числа n ($1 \leq n \leq 1000$), g и r ($1 \leq g, r \leq 10^9$) — Количество автомобилей, время зелёного и красного светов соответственно, в секундах. Далее, во второй строке задано n целых чисел a_i ($0 \leq a_i \leq g$) — сколько секунд необходимо каждому водителю, чтобы осознать, что уже можно ехать.

Формат выходного файла

Выведите единственное целое число — время, через которое уедет последний автомобиль.

Примеры

<code>trafficjam.in</code>	<code>trafficjam.out</code>
4 10 5 10 4 4 7	37

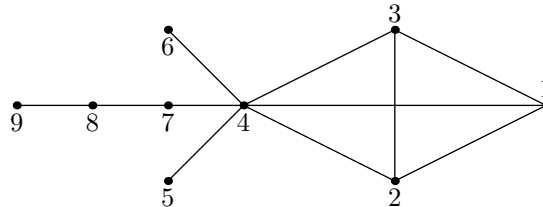
Примечание

В примере первый автомобиль проедет на 10-ой секунде, второй - на 19-ой секунде, третий - на 23-ей секунде и последний - на 37-ой.

Задача В. Хвост графа

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

Петя с детства любит воздушных змеев. Ему очень нравится наблюдать, как они летают. Особенно Пете нравится смотреть, как развевается хвост змея. На одном из уроков математики, которую Петя тоже любит, учитель рассказывал про графы. Во время рассказа, в качестве примера, учитель на доске нарисовал граф, похожий на воздушного змея. Пете сразу обратил внимание на *хвост* графа.



Хвостом графа назовем последовательность связанных вершин, такую что первая связана только со второй, вторая только с первой и третьей, а последующие только с соседними. Последняя вершина может быть связана либо только с предпоследней, либо с предпоследней и вершиной, которая не входит в наш хвост.

Теперь у Пети новое хобби — находить у графа самый длинный хвост, где длина хвоста определяется как количество вершин входящих в него. Но пока Петя не всегда может узнавать длину хвоста. Помогите ему.

Формат входного файла

В первой строке входного файла заданы числа n ($1 \leq n \leq 100000$) — количество вершин в графе и m ($1 \leq m \leq 200000$) — количество ребер в графе. Следующие m строк содержат по два числа a_i, b_i — номера вершин, которые соединяет соответствующее ребро.

Каждая пара вершин соединена не более чем одним ребром, никакое ребро не соединяет вершину с ней же. Из любой вершины существует путь до любой другой вершины графа.

Формат выходного файла

В выходной файл требуется вывести одно целое число — длину самого длинного хвоста графа.

Примеры

<code>graphtail.in</code>	<code>graphtail.out</code>
9 11 1 2 1 3 1 4 2 3 2 4 3 4 4 5 4 6 4 7 7 8 8 9	3

Примечание

Решения работающие для $n \leq 100$ и $m \leq 1000$ будут оцениваться из 40 баллов.

Задача С. Выборы

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

В стране Виртландии раз в пять лет проходят выборы президента. И каждый раз после завершения голосования появляется необходимость подсчитать результаты. Рассмотрим процесс обработки бюллетеней в Виртландии.

Всего в Виртландии голосуют n человек, соответственно необходимо обработать n бюллетеней, которые пронумерованы от 1 до n . Обработка длится k дней. Каждый день члены избирательной комиссии выбирают какое-то натуральное число, после чего обрабатывают все еще не обработанные бюллетени с номерами, которые делятся нацело на выбранное число. Так, в день с номером i обрабатываются все бюллетени, номера которых нацело делятся на a_i . После завершения процесса выяснилось, что подсчитаны были не все голоса. И теперь Ваша задача — выяснить, сколько бюллетеней осталось необработанными.

Формат входного файла

В первой строке входного файла даны два целых числа n и k ($1 \leq n \leq 300\,000$, $1 \leq k \leq 100\,000$) — количество голосующих граждан и количество дней для подсчета голосов, соответственно. Во второй строке даны k разделенных пробелами целых чисел a_i ($2 \leq a_i \leq n$) — числа, которые выбирались избирательной комиссией в каждый из дней.

Формат выходного файла

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

Примеры

<code>elections.in</code>	<code>elections.out</code>
8 2 2 3	3

Примечание

Решения, работающие при $1 \leq n \leq 10^4$ и $1 \leq k \leq 10^3$, будут оцениваться из 50 баллов.

В примере необработанными остались бюллетени номер 1, 5 и 7.

Задача D. Наименьшее общее кратное

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

Вася — юный программист. Он уже умеет решать задачи на графы, структуры данных, комбинаторику и динамическое программирование. Но вот с теорией чисел он еще не знаком. Поэтому он стал посещать лекции по теории чисел. На одном из первых занятий он узнал, что такое наименьшее общее кратное.

Набором чисел назовем множество, в котором элементы могут повторяться.

Наименьшее общее кратное набора S целых положительных чисел — минимальное положительное целое число, которое делится на каждый элемент набора S .

После каждого занятия преподаватель задает домашнее задание по пройденным темам. Вот и в этот раз он задал непростое задание:

Заданы числа n и k , требуется определить количество таких наборов из k элементов, наименьшее общее кратное которых равно n

Например, если $n = 6$ и $k = 2$, то подходящие наборы это: $\{1, 6\}$, $\{2, 3\}$, $\{2, 6\}$, $\{3, 6\}$, $\{6, 6\}$. Поэтому ответ будет равен 5. Заметим, что наборы, отличающиеся только порядком элементов, считаются одинаковыми.

Вася очень хорошо усвоил лекцию, а также он очень смысленный мальчик, поэтому он уже решил задачу. Но он не уверен в том, что все сделал правильно. Вася просит вас помочь проверить его программу: решите эту же задачу и найдите ответы на некоторые тесты. Так как ответ может быть очень большим, Вася решил, что ему достаточно буде знать остаток от деления ответа на $10^9 + 7$.

Формат входного файла

В первой строке входного файла через пробел заданы числа n ($1 \leq n \leq 10^{12}$) и k ($1 \leq k \leq 100$).

Формат выходного файла

В выходной файл требуется вывести одно число: ответ по модулю $10^9 + 7$.

Примеры

<code>lcm.in</code>	<code>lcm.out</code>
6 2	5
239 3	3

Примечание

Решения работающие для $n \leq 10^9$ и $k \leq 20$ будут оцениваться из 50 баллов.