

Задача А. Плохой массив

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

Вы нашли массив a из n элементов, и он вам не очень понравился, так что вы решили сделать его лучше. А именно - вы решили удалить из него как можно меньше элементов, так чтобы среди оставшихся элементов, для любых a_i и a_j выполнялось:

a_i делится на a_j или a_j делится на a_i

Например из массива $a = [2, 14, 42]$ не нужно удалять ни одного элемента

А из массива $a = [45, 9, 3, 18]$ придется удалить элемент 45 или элемент 18.

Так как массив достаточно большой, вы решили написать программу, которая за вас посчитает, сколько элементов из массива нужно удалить, чтобы он стал лучше.

Формат входных данных

В первой строке содержится одно целое число t ($1 \leq t \leq 10$) — количество наборов входных данных. Далее следуют t наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 2 * 10^5$) — длина массива a .

Вторая строка каждого набора входных данных содержит n чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 2 * 10^5$) — элементы массива a .

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — минимальное число элементов массива, которые необходимо удалить.

Примеры

стандартный ввод	стандартный вывод
1 5 2 6 12 6 3	1
2 7 3 5 15 9 18 25 30 6 5 6 7 8 9 10	4 4

Задача В. Очень большая сумма

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Вам дана бесконечная таблица **A**. Строки и столбцы в ней нумеруются с нуля. $A_{i,j} = 2^i \times 3^j$. Вам нужно обработать q запросов в формате i_1, i_2, j_1, j_2 , где ответом на запрос является

$$\left[\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} A_{i,j} \right] \bmod (10^9 + 7)$$

Формат входных данных

В первой строке входных данных записано число $1 \leq q \leq 10^5$. На следующих q строках записаны запросы в формате i_1, i_2, j_1, j_2 .

$$0 \leq i_1 \leq i_2 \leq 10^9$$

$$0 \leq j_1 \leq j_2 \leq 10^9$$

Формат выходных данных

Выведите q чисел — ответы на запросы.

Примеры

стандартный ввод	стандартный вывод
1 0 1 0 1	12
2 1 3 4 5 57 179 57 179	4536 140970084

Задача С. 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, \dots, i_k сходимой математиков, если $i_1 < i_2 < i_3 < \dots < i_k$ и $\gcd(a_{i_1}, a_{i_2}, \dots, a_{i_k}) > 1$. Эффективность этой сходимки равна $k \cdot \gcd(a_{i_1}, a_{i_2}, \dots, a_{i_k})$.

Найдите сумму эффективностей всех сходимок 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 ($1 \leq n \leq 200000$) — количество математиков.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 1000000$) — 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 сходимки — $1, 2, 1, 2$, так что ответ $1 \cdot 3 + 1 \cdot 3 + 2 \cdot 3 = 12$

Задача D. Расширенный алгоритм Евклида

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

Даны натуральные числа a, b, c . Если уравнение $ax + by = c$ имеет решения в целых числах, то выведите через пробел $GCD(a, b)$, x и y (какое-нибудь решение). Если решения не существует, то выведите слово **Impossible**.

Формат входных данных

Входные данные – натуральные числа и не превышают по модулю 10^4 .

Формат выходных данных

Выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
1 2 3	1 3 0
10 6 8	2 -4 8
3 3 1	Impossible

Задача Е. Метро

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

В городе N строят метро. Вася, житель города N , хочет знать, сколько станций окажется недалеко от его дома. Помогите ему.

Город N отличается очень строгой планировкой улиц: каждая улица идёт либо строго с юга на север, либо строго с востока на запад; при этом расстояние между соседними параллельными улицами одинаково. Соответственно, в городе есть много перекрёстков, расположенных в вершинах квадратной сетки. По планам, первая линия метро будет прямой и будет иметь станции на каждом перекрёстке, через который она пройдёт. Вася считает, что станция находится недалеко от его дома, если расстояние по прямой от его дома до станции не превосходит некоторой фиксированной величины R .

Формат входных данных

Введём систему координат с осью x , направленной с востока на запад, и осью y , направленной с юга на север, с началом координат на одном из перекрёстков и с единицей длины, равной расстоянию между соседними параллельными улицами. Таким образом, улицы будут прямыми с уравнениями $\dots, x = -2, x = -1, x = 0, x = 1, x = 2, \dots$, а также $\dots, y = -2, y = -1, y = 0, y = 1, y = 2, \dots$.

Во первой строке входного файла находятся целые числа x_0, y_0 — координаты Васиного дома (считаем, что он находится на некотором перекрёстке), — и расстояние R в тех же единицах измерения, в которых введены координаты. Во второй строке находятся четыре числа x_1, y_1, x_2, y_2 — координаты некоторых двух различных перекрёстков, через которые пройдёт линия метро. Все координаты во входном файле не превосходят 10^8 по модулю; расстояние R целое, положительное и не превосходит 10^8 .

Можете считать, что линия метро будет бесконечной в обоих направлениях.

Формат выходных данных

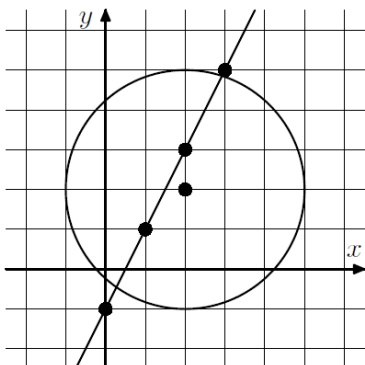
В выходной файл выведите одно число — количество станций, расположенных недалеко от Васиного дома.

Примеры

стандартный ввод	стандартный вывод
2 2 3 0 -1 1 1	2
0 0 1 -5 0 -3 0	3

Замечание

Первый пример соответствует рисунку; на рисунке дом Васи и станции метро обозначены жирными точками.



Задача F. Числа Каталана

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

Числа Каталана определяются следующим образом:

1. $C_0 = 1$

2. $C_n = \sum_{i=0}^{n-1} C_i C_{n-i-1}$

Ваша задача — посчитать $C_n \bmod m$.

Формат входных данных

На первой строке целые числа n ($0 \leq n \leq 1000$) и m ($1 \leq m \leq 10^9$).

Формат выходных данных

Выведите одно целое число — $C_n \bmod m$.

Примеры

стандартный ввод	стандартный вывод
5 1000000000	42
1 1000000000	1
515 1000000000	95405150

Задача G. Обратное по модулю

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

Даны два целых числа — a, m ($0 \leq a < m$).

Нужно найти такое целое x , что $ax \equiv 1 \pmod m$

Формат входных данных

На первой строке два целых числа — a, m ($0 \leq a < m \leq 10^{18}$).

Формат выходных данных

Если такого x не существует, выведите -1 . Иначе выведите целое x ($0 \leq x < m$). Если ответов несколько, выведите любой.

Примеры

стандартный ввод	стандартный вывод
7 30	13
179 817	639

Задача Н. Переедание

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

Однажды бабушка изготовила N печенек. Мы с сестрой собрались их сразу съесть, но бабушка кроме печенек изготовила и специальную инструкцию.

- Чтобы печенки не испортились, их нужно съесть за D дней.
- Будьте аккуратны и не переedayте! Каждый день нужно есть строго меньше X печенек.

И тут моя сестра говорит «Сколько же способов съесть печенки? Давай посчитаем!».

Два способа съесть печенки считаются различными, если есть такой день, что количество печенек, съеденных в этот день различаются в этих способах. Например, если N , D и X равны 5, 2 и 5 соответственно, количество способов равно 4:

- Съесть 1 печенку в первый день и оставшиеся 4 во второй.
- Съесть 2 печенки в первый день и оставшиеся 3 во второй.
- Съесть 3 печенки в первый день и оставшиеся 2 во второй.
- Съесть 4 печенки в первый день и оставшуюся 1 во второй.

Поскольку число способов может быть ужасно велико, я не хочу вместе с сестрой сидеть и страдать, я собираюсь написать программу, которая сама посчитает количество способов.

Формат входных данных

Входные данные состоят из одного или нескольких (не более 50) тестовых наборов. Для каждого тестового набора даны три числа на одной строке — N ($1 \leq N \leq 2000$), D ($1 \leq D \leq 10^{12}$) и X ($1 \leq X \leq 2000$). Конец ввода обозначается тремя нулями.

Формат выходных данных

Для каждого тестового набора выведите требуемое количество способов по модулю 1 000 000 007 на отдельной строке.

Пример

стандартный ввод	стандартный вывод
5 2 5	4
3 3 3	7
5 4 5	52
4 1 2	0
1 5 1	0
1250 50 50	563144298
0 0 0	

Задача I. Простая задача

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

Найдите количество натуральных чисел на данном отрезке от a до b включительно, не делящихся нацело ни на одно из заданных различных простых чисел p_i .

Формат входных данных

В первой строке входного файла заданы два числа a и b — границы отрезка ($1 \leq a \leq b \leq 10^{18}$). Во второй строке задано количество простых чисел n ($1 \leq n \leq 9$). В третьей строке перечислены сами простые числа p_i . Все числа p_i различны и не превосходят 100.

Формат выходных данных

Ответ должен содержать единственное целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5 10 2 2 3	2
20 40 2 3 7	12

Задача J. Старая книга

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

Группа юных археологов работает на раскопках здания древней библиотеки. Летом они обнаружили остатки старой книги и, изучив их, сделали следующие выводы: Книга содержит несколько страниц, каждая страница содержит либо текст, либо иллюстрацию. Первые k страниц книги точно содержат иллюстрации. Все страницы книги пронумерованы, но номер страницы написан только на страницах, содержащих текст. Сумма номеров страниц с текстом равна s . К сожалению, ни общее количество страниц в книге, ни количество иллюстраций установить не удалось. Тем не менее, юных археологов заинтересовал вопрос, какое минимальное количество иллюстраций могло быть в книге. Например, если $k = 1$, а $s = 8$, то страницы книги могли иметь следующее содержание (буквой «Т» обозначена страница, содержащая текст, а буквой «И» — страница, содержащая иллюстрацию):

- И Т И И И Т, пронумерованы страницы 2 и 6, всего 4 иллюстрации
- И И Т И Т, пронумерованы страницы 3 и 5, всего 3 иллюстрации
- И И И И И И И Т, пронумерована страница 8, всего 7 иллюстраций.

Минимальное количество иллюстраций равно 3. Требуется написать программу, которая по заданным целым числам k и s определяет минимальное количество иллюстраций, которое могло быть в книге

Формат входных данных

Первая строка входных данных содержит целое число k ($0 \leq k \leq 10^9$).

Вторая строка входных данных содержит целое число s ($k + 1 \leq s \leq 10^{12}$).

Формат выходных данных

Требуется вывести одно целое число — минимальное количество иллюстраций в книге.

Примеры

стандартный ввод	стандартный вывод
0 10	0
0 11	1
0 12	1