

Задача А. Решето Эратосфена

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

По введенным числам A и B вывести все простые числа в интервале от A до B включительно.

Формат входных данных

В единственной строке вводятся два числа $1 \leq A \leq B \leq 1000000$

Формат выходных данных

Вывести в одну строку все простые числа в интервале от A до B включительно

Примеры

<code>stdin</code>	<code>stdout</code>
2 2	2
1 100	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Задача В. Возведение в степень

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Формат входных данных

Во входном файле даны три натуральных числа A , B , M ($1 \leq A, B \leq 10^9, 2 \leq M \leq 10^9$), записанные на одной строке через пробел.

Формат выходных данных

В выходной файл выведите одно число, равное $A^B \bmod M$ (\bmod означает взятие остатка при делении).

Примеры

<code>stdin</code>	<code>stdout</code>
2 3 100	8

Замечание

При решении данной задачи нельзя использовать встроенную операцию возведения в степень.

Задача С. Битовые операции

Имя входного файла: bits.in
Имя выходного файла: bits.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В данной задаче даны целые числа A и k . В выходной файл, по одному в строке, выведите целые числа, являющиеся ответами на следующие запросы:

1. Значение k -го бита числа A , т.е 0 или 1
2. Число, которое получается из A установкой k -го бита, равным 1
3. Число, которое получается из A установкой k -го бита, равным 0
4. Число, которое получается из A инвертированием k -го бита
5. Число, которое получается из A , если обнулить k его последних бит.
6. Число, которое состоит из последних k бит числа A

Формат входных данных

Во входном файле записаны через пробел два числа — A ($0 \leq A < 2^{31}$) и k ($0 \leq k \leq 31$).

Формат выходных данных

В выходной файл выведите ответы на запросы, указанные в условии задачи.

Примеры

bits.in	bits.out
5 1	0 7 5 7 4 1
29 3	1 29 21 21 24 5

Замечание

Запрещается использовать циклы, условия, сложение, вычитание (кроме получения константы -1), умножение и деление.

Биты нумеруются с нуля, то есть младший бит имеет номер 0.

Задача D. Обратный элемент по простому модулю

Имя входного файла: `inv.in`
Имя выходного файла: `inv.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Обратным элементом к a в кольце вычетов по модулю p называется такой элемент x , что выполняется равенство $ax \equiv 1 \pmod{p}$.

Формат входных данных

Входной файл содержит два целых числа a и p ($1 \leq a, p \leq 10^9$), причем p — простое.

Формат выходных данных

В выходной файл выведите обратный элемент к a в кольце вычетов по модулю p .

Примеры

<code>inv.in</code>	<code>inv.out</code>
1 2	1

Задача Е. Часовая башня

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Многие ученики Хогвартса разозлились на то, что Амбридж запретила собираться в группы из трёх и более учеников, и решили придумать, как ей насолить. Один из планов состоял в том, чтобы прийти в Часовую башню и сломать имеющиеся там часы.

Часы состоят из N последовательно соединённых шестерней, то есть первая шестерня зацепляет вторую, вторая зацепляет третью и так далее... $(N - 1)$ -я шестерня зацепляет последнюю N -ю.

Перед тем как сломать часы, ученики сначала хотят разобраться, как они работают. Им интересно, сколько раз нужно полностью повернуть первую шестерню, чтобы все остальные шестерни вернулись в изначальное положение.

Формат входных данных

В первой строке вводится целое число N ($2 \leq N \leq 10$). В следующей строке вводится N целых чисел a_i — число зубчиков в каждой шестерне ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите минимальное число полных поворотов первой шестерни, после которой все остальные шестерни тоже возвращаются в изначальное положение. Гарантируется, что произведение первого числа и ответа не превышает 10^9 .

Примеры

стандартный ввод	стандартный вывод
3 10 20 30	6

Задача F. Факториал

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Факториалом $N!$ числа называется произведение всех натуральных чисел от 1 до N . Также, для удобства, считается, что $0! = 1$. В этой задаче от вас требуется представить число $N!$ в виде произведения $p_1^{q_1} * p_2^{q_2} * \dots * p_m^{q_m}$, где p_i - различные простые, а q_i - положительные целые числа.

Формат входных данных

На вход подаётся единственное целое число N ($1 \leq N \leq 1\,000\,000$), факториал которого необходимо разложить на множители.

Формат выходных данных

В первой строке нужно вывести количество множителей в разложении. В остальных — множители, по одному в строке в порядке возрастания основания. При этом в каждой строке нужно через пробел указать основание и положительную степень каждого множителя.

Примеры

stdin	stdout
6	3 2 4 3 2 5 1

Задача G. НОД

Имя входного файла: `sumgcd.in`
Имя выходного файла: `sumgcd.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Коля очень любит число N . Однажды ему захотелось разложить число N на сумму двух натуральных чисел, $N = A + B$. Поскольку эта крайне простая задача, он захотел разложить число так, чтобы максимизировать наибольший общий делитель A и B . Эту сложную задачу Коля решить уже не может, поэтому обращается к вам за помощью.

Формат входных данных

В единственной строке входного файла записано число N ($2 \leq N \leq 10^9$)

Формат выходных данных

Нужно вывести два числа A и B , такие, чтобы $\text{НОД}(A, B)$ был максимально возможным, а их сумма равна N

Примеры

	<code>sumgcd.in</code>	<code>sumgcd.out</code>
	15	5 10
	16	8 8