

## Задача А. Разложение на множители

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

64 мегабайта

Дано число. Требуется разложить его на простые множители.

### Формат входных данных

Вводится число  $N$  ( $2 \leq N \leq 10^9$ ).

### Формат выходных данных

Выведите через пробел разложение на простые множители в порядке возрастания множителей.

### Примеры

стандартный ввод	стандартный вывод
17	17
60	2 2 3 5

## Задача В. Диофантово уравнение

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

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

Входной файл содержит три целых числа  $a$  и  $b$  и  $c$  ( $1 \leq a, b, c \leq 10^4$ ).

### Формат выходных данных

В выходной файл выведите искомые  $x$  и  $y$  через пробел. Если решения не существует, выведите одну строку «Impossible».

### Пример

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

## Задача С. Система линейных сравнений

Имя входного файла: `chinese.in`  
Имя выходного файла: `chinese.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

256 мегабайт

Дана система из двух линейных сравнений:

$$\begin{cases} x \equiv a \pmod{n}, \\ x \equiv b \pmod{m}; \end{cases}$$

где числа  $n$  и  $m$  не обязательно взаимно простые. Решите эту систему или определите, что она не имеет решений.

### Формат входных данных

В первой строке входного файла записано единственное число  $1 \leq t \leq 100\,000$ . В следующих  $t$  строках содержатся по четыре целых числа  $a, b, n, m$ , задающих одну систему сравнений. Все числа не превосходят по модулю  $10^4$ ,  $n > 1$ ,  $m > 1$ .

### Формат выходных данных

Программа должна вывести  $t$  строк, по одной на каждую систему.

В случае, если система не имеет решений, выведите строку "NO".

В случае, если решение есть, то необходимо вывести слово "YES" и два таких числа  $x_0$  и  $p$ ,  $0 \leq x_0 < p$ , такие, что множество чисел  $x = x_0 + kp$ , где  $k$  — произвольное целое число является решением данной системы.

### Пример

<code>chinese.in</code>	<code>chinese.out</code>
3	YES 38 45
3 2 5 9	YES 1 45
1 1 5 9	NO
7 13 20 24	

## Задача D. Обратное по модулю

Имя входного файла: `inverse2.in`  
Имя выходного файла: `inverse2.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Даны два целых числа —  $a, m$  ( $0 \leq a < m$ ). Нужно найти такое целое  $x$ , что  $a \cdot x \equiv 1 \pmod{m}$ .

### 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.

### Пример

<code>inverse2.in</code>	<code>inverse2.out</code>
7 30	13

## Задача Е. Решето Эратосфена

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
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

стандартный ввод	стандартный вывод
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 F. Шарики

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

У меня есть несколько (скажем,  $n$ ) шариков (маленьких стеклянных шариков), и я собираюсь купить несколько коробок для их хранения. Коробки бывают двух типов:

- каждая коробка стоит  $c_1$  и может содержать ровно  $n_1$  шариков
- каждая коробка стоит  $c_2$  и может содержать ровно  $n_2$  шариков

Я хочу, чтобы каждая из использованных коробок была заполнена до отказа, а также чтобы свести к минимуму общую стоимость их покупки. Поскольку мне трудно понять, как распределить шарик по ящикам, я обращаюсь к вам за помощью. Я хочу, чтобы ваша программа также была эффективной.

### Формат входных данных

Входной файл может содержать несколько тестов. Каждый тест начинается со строки, содержащей целое число  $n$  ( $1 \leq n \leq 2,000,000,000$ ). Вторая строка содержит  $c_1$  и  $n_1$ , а третья строка содержит  $c_2$  и  $n_2$  здесь  $c_1, c_2, n_1$  и  $n_2$  — положительные целые числа, имеющие значения меньше 2,000,000,000. Ввод завершается числом 0

### Формат выходных данных

Для каждого теста выведите строку, содержащую решение с минимальной стоимостью (два неотрицательных целых числа  $m_1$  и  $m_2$ , где  $m_i$  — количество требуемых коробок  $i$ -го типа. Если решения не существует, выведите "failed". Если решений несколько, выведите любое.

### Примеры

стандартный ввод	стандартный вывод
43 1 3 2 4 0	13 1
40 5 9 5 12 0	failed

## Задача G. Сигма-функция на отрезке

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

Нужно научиться считать  $\sum_{i=L}^R \sigma(i)$ . Где  $\sigma(n)$  — сумма натуральных делителей числа  $n$ .

### Формат входных данных

Последовательность из не более чем  $10^5$  запросов. Каждый запрос записан на отдельной строке. Формат запроса прост: числа  $L, R$  ( $1 \leq L \leq R \leq 5 \cdot 10^6$ ).

### Формат выходных данных

Для каждого запроса нужно вывести одно число —  $\sum_{i=L}^R \sigma(i)$ .

### Пример

<code>lrsigma.in</code>	<code>lrsigma.out</code>
3 10	83
3 3	4
10 10	18

## Задача Н. Больше простых!

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 15 секунд  
Ограничение по памяти: 64 мегабайта

Найдите все простые числа не большие  $n$ . Поскольку  $n$  в этой задаче не просто большое, а прямо здоровенное, для того чтобы проверить, что вы нашли числа правильно, мы попросим вас посчитать от найденных чисел специальный хеш.

Хеш будет считаться по следующему алгоритму. В начале переменная  $h = 0$ . После каждого очередного встреченного простого числа  $p_i$ , будем пересчитывать  $h$  по формуле  $h = h \cdot x + p_i$ , при этом будем игнорировать переполнение знакового 32-битного целого типа. Значение переменной  $h$  в конце — это хеш, который вам нужно вывести.

### Формат входных данных

Входной файл содержит два числа  $n$  ( $2 \leq n \leq 10^9$ ) и  $x$  ( $1 \leq x \leq 10^9$ ).

### Формат выходных данных

Выведите полученный хеш.

### Примеры

стандартный ввод	стандартный вывод
10 10	2357
11 100	203050711
1000000000 2	1576840463



## Задача I. Видеокарты

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

Маленький Влад обожает играть в популярную игру «Vota-2» на своем компьютере. На днях разработчики сообщили о выходе её продолжения под названием «Vota-3». Разумеется, Влад тут же её приобрёл, но оказалось, что компьютер Влада слишком старый для этой игры, поэтому он отправился в магазин за обновлением для него.

В магазине Влад увидел  $n$  видеокарт,  $i$ -я из которых имеет свою мощность  $a_i$ , выражаемую целым положительным числом. Для того, чтобы игра точно заработала, Влад решил купить не одну, а несколько видеокарт и объединить их мощности по новейшей технологии. Для использования этой технологии одна из видеокарт выбирается в качестве ведущей, а все остальные видеокарты подключаются к ней в качестве вторичных. Для корректной работы этой конструкции необходимо, чтобы мощность каждой видеокарты была кратна мощности ведущей видеокарты. Для обеспечения совместимости мощность каждой вторичной видеокарты может быть понижена до любого меньшего целого положительного значения. При этом мощность опорной видеокарты должна оставаться оригинальной, то есть не может быть уменьшена.

У Влада очень много денег, поэтому он может купить любые видеокарты. Помогите ему выбрать такие видеокарты, чтобы после необходимого понижения их мощностей и соединения их по новой технологии суммарная итоговая мощность видеокарт была максимальной.

### Формат входных данных

В первой строке находится целое число  $n$  ( $1 \leq n \leq 200\,000$ ) – количество видеокарт в магазине.

Во второй строке находится  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 200\,000$ ), обозначающих мощности видеокарт.

### Формат выходных данных

В единственной строке выведите целое число – максимальную возможную суммарную итоговую мощность видеокарт.

### Примеры

стандартный ввод	стандартный вывод
4 3 2 15 9	27
4 8 2 2 7	18