

## Задача А. Первообразный корень по простому модулю

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

Дано простое нечётное число  $p$ . Требуется найти минимальный первообразный корень по модулю  $p$ .

### Формат входных данных

Первая строка входного файла содержит простое число  $p$  ( $3 \leq p \leq 10^9$ ).

### Формат выходных данных

Выведите ответ на поставленную задачу.

### Примеры

<code>primroot.in</code>	<code>primroot.out</code>
3	2
239	7
127	3

## Задача В. Пётя

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 4.5 секунд  
Ограничение по памяти: 512 мегабайт

Петя хочет посчитать значения  $\phi(1), \phi(2), \dots, \phi(n)$ . И вы должны ему помочь в этом.

Он опаздывает на поезд в Петербург, так что лимит по времени в этой задаче поставлен примерно впритык. Также в этой задаче маленький лимит по памяти, потому что ноутбук Петра очень старый и не может выделить на решение задачи больше чем данное количество памяти.

### Формат входных данных

Число  $n$  ( $1 \leq n \leq 10^8$ ).

### 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

## Задача С. Больше простых)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 10 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

Два числа —  $n$  ( $1 \leq n \leq 10^9$ ) 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
10000000 2	-538886859

## Задача D. Дискретное логарифмирование

Имя входного файла: `logging.in`  
Имя выходного файла: `logging.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дано простое число  $P$  и целые числа  $B$  и  $N$ , лежащие в пределах от 2 до  $P - 1$ . Вычислите дискретный логарифм числа  $N$  по основанию  $B$  по модулю  $P$ , то есть наименьшее целое неотрицательное  $L$ , для которого  $B^L \equiv N \pmod{P}$ .

### Формат входных данных

Ввод состоит из нескольких строк, в каждой из которых находятся простое число  $P$  и целые числа  $B$  и  $N$ , разделённые пробелом ( $2 \leq B, N < P < 2^{31}$ ).

### Формат выходных данных

Для каждой строки ввода выведите дискретный логарифм в отдельной строке. Если дискретного логарифма для теста не существует, в соответствующей строке выведите «no solution».

### Примеры

<code>logging.in</code>	<code>logging.out</code>
5 2 3	3
1000000007 4 256	4
1000000007 4 512	250000006
19 4 18	no solution
19 18 4	no solution

## Задача Е. Ну Поллард, и Поллард...

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

Вам необходимо разложить число  $n$  на простые множители. Сегодня ночью вам приснился Джон Поллард и сообщил вам половину простых делителей числа  $n$ . Формально, если  $n = p_0 \times \dots \times p_k$ , где  $p_0 \leq \dots \leq p_k$ , то Поллард сообщил вам числа  $p_0, p_2, \dots$ .

Пользуясь информацией, которую сообщил вам Поллард, найти полную факторизацию числа.

### Формат входных данных

Первая строка входа содержит два целых числа — число  $n$  и  $l$  — количество делителей, которые вам сообщил Поллард ( $2 \leq n \leq 10^{18}$ ). Вторая строка содержит  $l$  простых чисел —  $p_0, p_2, \dots$ .

### Формат выходных данных

Выведите простые делители  $n$  в порядке неубывания. Количество выводить не нужно.

### Примеры

<code>factor.in</code>	<code>factor.out</code>
12 2 2 3	2 2 3
7 1 7	7
1764 3 2 3 7	2 2 3 3 7 7

## Задача F. Проверка на простоту

Имя входного файла: `prime.in`  
Имя выходного файла: `prime.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Проверьте, являются ли данные числа простыми.

### Формат входных данных

Программа получает на вход одно целое число  $k$ ,  $1 \leq k \leq 100$ . Следующие  $k$  строк содержат по одному числу  $n_i$ ,  $2 \leq n_i \leq 10^{18}$ .

### Формат выходных данных

Для каждого из чисел  $n_i$  программа должна вывести одну строку: **YES** для простого числа и **NO** для составного.

### Примеры

<code>prime.in</code>	<code>prime.out</code>
2	YES
3	NO
4	

## Задача G. Факторизация

Имя входного файла: pollard.in  
Имя выходного файла: pollard.out  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Дано натуральное число. Факторизуйте его, то есть представьте в виде произведения набора простых чисел. Число  $p$  называется простым, если имеет ровно два различных натуральных делителя: 1 и  $p$ .

### Формат входных данных

В единственной строке записано единственное натуральное число  $N$ .  $2 \leq N \leq 9 \cdot 10^{18}$ .

### Формат выходных данных

Выведите в неубывающем порядке одно или несколько простых чисел, произведение которых равно  $N$ .

### Примеры

pollard.in	pollard.out
6	2 3
7	7