

## Задача А. Вычислите функции

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

Дано число  $N$ . Требуется вычислить следующие функции для него:

$\varphi(N)$  = количество взаимно простых с  $N$  чисел среди  $1, 2, \dots, N$

$\tau(N)$  = количество делителей числа  $N$

$\sigma(N)$  = сумма всех делителей числа  $N$

### Формат входного файла

Во входном файле содержится единственное число  $1 \leq N \leq 10^9$ .

### Формат выходного файла

В единственную строку выходного файла выведите через пробел три числа — значения  $\varphi(N)$ ,  $\tau(N)$ ,  $\sigma(N)$ .

### Примеры

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

## Задача В. Асимптотический закон возвращается

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

Как и в предыдущей задаче, вам дается единственное число  $N$ . В этот раз требуется посчитать количество простых чисел среди  $1, 2, \dots, N$ .

### Формат входного файла

Во входном файле содержится единственное число  $1 \leq N \leq 25\,000\,000$ .

### Формат выходного файла

В выходной файл выведите искомое количество.

### Примеры

<code>asyp.in</code>	<code>asyp.out</code>
10	4

## Задача С. Хорошие числа

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

Один мальчик Коля разобрался с простыми числами и решил придумать свою задачу. Для нее ему потребовалось найти сумму всех чисел в заданном интервале, которые обладали бы «хорошим» свойством. Хорошие числа определяются следующим образом. На основе числа — кандидата на «хорошесть» строим последовательность. Первым элементом в ней является само это число, вторым — количество всех его различных делителей. Третий элемент — количество всех делителей второго элемента и так далее. Назовем число «хорошим», если полученная последовательность не содержит полных квадратов целых чисел, больших 1.

### Формат входного файла

Во входном файле записаны два натуральных числа  $A$  и  $B$  ( $1 \leq A < B \leq 10^7$ ).

### Формат выходного файла

В выходной файл нужно выдать сумму всех «хороших» чисел из заданного интервала, включая границы.

### Примеры

<code>good.in</code>	<code>good.out</code>
5 8	12

### Примечание

В примере «хорошими» являются числа 5 и 7.

## Задача D. Снова простые числа

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

Решая задачу В, Коля дошел до крайности. После долгой и кропотливой работы компьютера в течение всей ночи ему осталось проверить на простоту всего несколько десятков тысяч чисел. Однако ни Коля, ни его компьютер продолжать не в состоянии, поэтому ответственность за проверку их ложится на вас.

### Формат входного файла

В первой строке входного файла содержится единственное число  $1 \leq t \leq 20000$  — количество чисел, которые осталось проверить на простоту. В следующих  $t$  строках по одному на каждой строке записано  $t$  целых положительных чисел, не превосходящих  $10^9$ .

### Формат выходного файла

В  $i$ -й строке выходного файла должно быть записано «YES», если  $i$ -е число простое, и «NO» иначе.

### Примеры

again.in	again.out
2	YES
3	NO
4	

### Задача E. Длинный НОД

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

### Формат входного файла

Входной файл содержит два натуральных числа  $a$  и  $b$  ( $1 \leq a, b \leq 10^{500}$ ).

### Формат выходного файла

В выходной файл выведите наибольший общий делитель этих двух чисел.

### Примеры

longgcd.in	longgcd.out
6 9	3
89035489 801032	1

### Задача F. RSA. Взлом RSA

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

В 1977 году Ronald Linn Rivest, Adi Shamir и Leonard Adleman предложили новую криптографическую схему RSA, используемую до сих пор. RSA является криптосистемой с открытым ключом: зашифровать сообщение может кто угодно, знающий общеизвестный открытый ключ, а расшифровать сообщение — только тот, кто знает специальный секретный ключ.

Желающий использовать систему RSA для получения сообщений должен сгенерировать два простых числа  $p$  и  $q$ , вычислить  $n = pq$  и сгенерировать два числа  $e$  и  $d$  такие, что  $ed \equiv 1 \pmod{(p-1)(q-1)}$  (заметим, что  $(p-1)(q-1) = \varphi(n)$ ). Числа  $n$  и  $e$  составляют открытый ключ и являются общеизвестными. Число  $d$  является секретным ключом, также

необходимо хранить в тайне и разложение числа  $n$  на простые множители, так как это позволяет вычислить секретный ключ  $d$ .

Сообщениями в системе RSA являются числа из  $\mathbb{Z}_n$ . Пусть  $M$  — исходное сообщение. Для его шифрования вычисляется значение  $C = M^e \pmod n$  (для этого необходимо только знание открытого ключа). Полученное зашифрованное сообщение  $C$  передается по каналу связи. Для его расшифровки необходимо вычислить значение  $M = C^d \pmod n$ , а для этого необходимо знание секретного ключа.

Вы перехватили зашифрованное сообщение  $C$  и знаете только открытый ключ: числа  $n$  и  $e$ . «Взломайте» RSA — расшифруйте сообщение на основе только этих данных.

### Формат входного файла

Программа получает на вход три натуральных числа:  $n$ ,  $e$ ,  $C$ ,  $n \leq 10^9$ ,  $e \leq 10^9$ ,  $C < n$ . Числа  $n$  и  $e$  являются частью какой-то реальной схемы RSA, т.е.  $n$  является произведением двух простых и  $e$  взаимно просто с  $\varphi(n)$ . Число  $C$  является результатом шифрования некоторого сообщения  $M$ .

### Формат выходного файла

Выведите одно число  $M$  ( $0 \leq M < n$ ), которое было зашифровано такой криптосхемой.

### Примеры

rsa.in	rsa.out
143 113 41	123
9173503 3 4051753	111111