

## Задача А. Количество делителей

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

Дано целое число  $n$ . Требуется найти количество его делителей, не считая 1 и само число  $n$ .

Программа должна содержать функцию

- `function CountDivisors(n: longint): longint` (Pascal),
- `int CountDivisors(int n)` (C++),
- `def CountDivisors(n)` (Python),

возвращающую количество делителей числа  $n$ . Программа должна считать число  $n$ , вызвать функцию `CountDivisors`, напечатать одну из двух строк в зависимости от результата.

### Формат входного файла

Программа получает на вход единственное натуральное число  $n$  ( $2 \leq n \leq 2^{31} - 1$ ).

### Формат выходного файла

Программа должна выводить одно число — искомое количество делителей.

### Примеры

	<code>divisors.in</code>	<code>divisors.out</code>
3		0
4		1

### Note

Программа, не использующая описанную функцию, засчитываться не будет.

## Задача В. Функция `printlist`

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

Напишите функцию `printlist`, которая получает на вход 4 аргумента:

- список `s`
- строку `before`: что печатать в начале списка
- строку `between`: что печатать между элементами списка
- строку `after`: что печатать после списка

и выводит на экран список `s` в указанном формате. Вставьте вашу функцию в начало следующей программы и сдайте ее на проверку:

```
s = input().split()
printlist(s, 'x=', '*', '.')
printlist(s, '', ' ', '')
printlist(s, '', '', '')
printlist(s, '"', '"', '"', '"')
```

Не забывайте, что список может состоять из 0 или 1 элемента!

### Формат входного файла

### Формат выходного файла

### Примеры

<code>stdin</code>	<code>stdout</code>
<code>abcd e fg</code>	<code>x=abcd*e*fg.</code> <code>abcd e fg</code> <code>abcdefg</code> <code>"abcd","e","fg"</code>

## Задача С. Разложение на простые множители

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

По введенному числу  $N$  выведите его разложение на простые множители.

### Формат входного файла

В единственной строке вводится единственное число  $2 \leq N \leq 100000$ .

### Формат выходного файла

Выведите в одной строке последовательность чисел  $a_i$  таких что их произведение равно  $N$  и при этом  $a_i \leq a_{i+1}$

### Примеры

<code>prime.in</code>	<code>prime.out</code>
24	2 2 2 3
2	2

### Задача D. Решето Эратосфена

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

### Note

В тесте 1 :  $24 = 2^3 * 3^1$   
В тесте 2 :  $2 = 2^1$

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

### Формат входного файла

В единственной строке вводятся два числа  $1 \leq A \leq B \leq 100000$

### Формат выходного файла

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

### Примеры

sieve.in	sieve.out
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 E. Разложение на простые множители - 2

Имя входного файла: prime2.in  
Имя выходного файла: prime2.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

По введенному числу  $N$  выведите его разложение на простые множители.

### Формат входного файла

В единственной строке вводится единственное число  $2 \leq N \leq 100000$ .

### Формат выходного файла

Выведите в каждой строке по два числа. Первое число — делитель, второе — степень, с которой он входит в разложение числа. Если степень равна 0, то выводить делитель и степень не надо. Пары чисел выводить по возрастанию делителя. Внимательно прочитайте примеры.

### Примеры

prime2.in	prime2.out
24	2 3 3 1
2	2 1