

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

возвращающую количество делителей числа n . Программа должна считать число n , вызвать функцию `count_divisors`, напечатать ответ.

Формат входных данных

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

Формат выходных данных

Программа должна выводить одно число — искомое количество делителей.

Примеры

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

Замечание

Программа, не использующая описанную функцию, засчитываться не будет.

Задача В. Решето Эратосфена

Имя входного файла: `sieve.in`
Имя выходного файла: `sieve.out`
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 $1 \leq A \leq B \leq 100000$

Формат выходных данных

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

Примеры

| <code>sieve.in</code> | <code>sieve.out</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 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 входных данных

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

Формат выходных данных

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. 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 входных данных

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

Формат выходных данных

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 |

Замечание

В тесте 1 : $24 = 2^3 * 3^1$

В тесте 2 : $2 = 2^1$

Задача Е. 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` и строки `before`, `between`, `after`. Эта функция должна выводить на экран список в следующем формате: в начале списка строка `before`, затем элементы списка, разделенные строкой `between`, в конце строка `after`.

Вставьте вашу функцию в начало приведенной ниже 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 входных данных

Вам не надо об этом задумываться.

Формат выходных данных

И об этом тоже.

Примеры

| stdin | stdout |
|-----------|---|
| abcd e fg | x=abcd*e*fg. abcd e fg abcdefg "abcd","e","fg" |

Задача F. Сортировка пузырьком-2

Имя входного файла: bubble2.in
Имя выходного файла: bubble2.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Определите, сколько обменов сделает алгоритм пузырьковой сортировки по возрастанию для данного массива.

Формат входных данных

На первой строке дано число N ($1 \leq N \leq 1000$) — количество элементов в массиве. На второй строке — сам массив. Гарантируется, что все элементы массива различны и не превышают по модулю 10^9 .

Формат выходных данных

Выведите одно число — количество обменов пузырьковой сортировки.

Примеры

| bubble2.in | bubble2.out |
|--------------|-------------|
| 3 1 3 2 | 1 |
| 2 2 1 | 1 |
| 4 4 1 5 3 | 3 |

Замечание

В отличие от предыдущего дня, **ОБЯЗАТЕЛЬНО**: Использование Функций