

Задача А. Массив по модулю

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

В этой задаче Вам нужно будет считать массив и, используя специальную процедуру, которой передается считанный массив, присвоить всем его элементам их же значения по модулю.

Примечание. Решения, которые не будут содержать требуемой процедуры, не будут засчитываться.

Формат входного файла

В первой строке входного файла записано натуральное число N ($N \leq 10^3$), размер массива. Во второй строке через пробел идут N целых чисел, по модулю не превосходящих 30000, — элементы массива.

Формат выходного файла

В выходной файл выведите преобразованный массив.

Примеры

<code>array.in</code>	<code>array.out</code>
5	2 6 1 9 13
2 -6 -1 9 13	

Задача В. Количество делителей

Имя входного файла: `divisors.in`
Имя выходного файла: `divisors.out`
Ограничение по времени: 1 секунда
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>
2	0
4	1

Задача С. Переворот

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

Дано натуральное число N и последовательность из N элементов. Требуется вывести эту последовательность в обратном порядке.

Примечание. В программе запрещается объявлять массивы и использовать циклы (даже для ввода).

Формат входного файла

В первой строке входного файла записано натуральное число N ($N \leq 10^3$). Во второй строке через пробел идут N целых чисел, по модулю не превосходящих 1000, — элементы последовательности.

Формат выходного файла

В выходной файл выведите заданную последовательность в обратном порядке.

Примеры

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

Задача D. МегаНОД возвращается

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

Найдите НОД N заданных чисел.

Примечание. В программе запрещается объявлять массивы и использовать циклы (даже для ввода).

Формат входного файла

Первая строка входного файла содержит натуральное число N ($1 \leq N \leq 1000$) — количество чисел. Во второй строке заданы N чисел, не превышающие по модулю 10^9 .

Формат выходного файла

Выведите НОД N чисел.

Пример

megagcd.in	megagcd.out
1 3	3
3 9 15 22	1

Задача E. Ханойские башни

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

Даны три колышка. На первом колышке находятся N дисков сверху вниз по возрастанию размера диска. Два другие пустые. Требуется перенести все диски с первого колышка на второй. Переносить диски разрешается только по одному. Не разрешается класть больший диск на меньший. Для предотвращения заикливания, как это произошло с буддийскими монахами, не используйте циклы в программе.

Формат входного файла

Дано целое число N ($N \in [1, 20]$).

Формат выходного файла

Выведите по два числа в строке — номер колышка, откуда и куда переносится диск. Выведите кратчайшее решение.

Примеры

hanoi.in	hanoi.out
1	1 2
2	1 3 1 2 3 2