

Во всех задачах необходимо реализовать те функции, о которых говорится в условии, а также их содержательно использовать (т.е. работа программы должна опираться на них). В том числе и в тех задачах, где заголовок функции не дан. Функция должна возвращать результат, а не выводить его. Соблюдение этих условий не проверяется автоматически и будет проверено вручную перед оцениванием..

Задача А. Шкаф

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Размеры шкафа $A \times B \times C$. Возможно ли его пронести через дверной проем размерами $X \times Y$? Запрещается проносить шкаф под углом, т.е. одна из граней шкафа должна быть параллельна плоскости пола.

Замечание

Реализуйте и используйте функцию, проверяющую можно ли пронести шкаф в заданной ориентации.

Формат входных данных

Даны целые числа A, B, C, X, Y ($1 \leq A, B, C, X, Y \leq 100$).

Формат выходных данных

Выходной файл должен содержать строку «YES», если пронести возможно, «NO» — если нельзя.

Примеры

ВВОД	ВЫВОД
4 5 6 10 20	YES
4 5 6 3 4	NO
6 5 4 4 5	YES

Задача В. Послезавтра

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

По заданной дате требуется определить, какое число будет послезавтра.

Замечание

Реализуйте и используйте функцию, определяющую является ли год високосным и функцию, прибавляющую один день к заданной дате. Високосными являются года кратные 4 и не кратные 100, кроме кратных 400.

Формат входных данных

Дано число, месяц и год (год — число в промежутке от 1 до 10000).

Формат выходных данных

Требуется вывести, какое число будет послезавтра, в формате входных данных.

Примеры

ВВОД	ВЫВОД
1 8 2009	3 8 2009
30 12 2009	1 1 2010
28 2 2008	1 3 2008

Задача С. Количество делителей

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

- `int CountDivisors(int n)` (C++),
- `def CountDivisors(n)` (Python),

возвращающую количество делителей числа n . Программа должна считать число n , вызвать функцию `CountDivisors`, 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

ВВОД	ВЫВОД
2	0
4	1

Задача D. Минимальный делитель числа

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Напишите функцию

- `int MinDivisor (int n)` (C/C++),
- `def MinDivisor(n)` (Python),

находящую наименьший делитель числа n , отличный от 1.

Формат входных данных

На вход дается единственное натуральное число n больше 1 и не превышающее $2^{31} - 1$.

Формат выходных данных

Выведите результат работы функции на числе n .

Примеры

ввод	вывод
4	2
5	5

Задача Е. Массив по модулю

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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

Формат входных данных

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 выходных данных

В выходной файл выведите преобразованный массив.

Примеры

ВВОД	ВЫВОД
5 2 -6 -1 9 13	2 6 1 9 13

Задача F. Двойной переворот

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дана последовательность натуральных чисел $1, 2, 3 \dots N$ ($1 \leq N \leq 1000$). Необходимо сначала расположить в обратном порядке часть этой последовательности от элемента с номером A до элемента с номером B , а затем от C до D ($A < B$; $C < D$; $1 \leq A, B, C, D \leq 1000$).

Реализуйте функцию, которая принимает в качестве параметра массив и выполняет один переворот.

Формат входных данных

Даны числа N, A, B, C, D .

Формат выходных данных

Требуется вывести полученную последовательность.

Примеры

ВВОД	ВЫВОД
9 2 5 6 9	1 5 4 3 2 9 8 7 6
9 3 6 5 8	1 2 6 5 8 7 3 4 9