

## Содержание

1	Задача А. Сумма двух	2
2	Задача В. Произведение двух	3
3	Задача С. Наибольший общий делитель	4
4	Задача D. Число Фибоначчи	5
5	Задача Е. Два числа	6
6	Задача F. Самая большая буква	7
7	Задача G. Сумма значений функции	8
8	Задача H. Очень большие числа	9
9	Задача I. Лишние пробелы	10
10	Задача J. Коробки	11
11	Задача K. Тестирующая система	12

# Непростые задачи

## 1 Задача А. Сумма двух

Имя входного файла: `sum.in`  
Имя выходного файла: `sum.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

### Формат входных данных

В первой строке входного файла расположены два целых числа  $A$  и  $B$ , не превосходящих 1 000 по модулю.

### Формат выходных данных

Ваша программа должна выдавать в выходной файл одно число — сумму чисел  $A$  и  $B$ .

### Примеры

<code>sum.in</code>	<code>sum.out</code>
2 3	5
17 -18	-1

## 2 Задача В. Произведение двух

Имя входного файла: `product.in`  
Имя выходного файла: `product.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

### Формат входных данных

Входной файл состоит из двух целых чисел  $A$  и  $B$ , не превосходящих по модулю  $10^9$ .

### Формат выходных данных

Программа должна выдавать в выходной файл единственное число — произведение чисел  $A$  и  $B$ .

### Пример

<code>product.in</code>	<code>product.out</code>
2 2	4

### 3 Задача С. Наибольший общий делитель

Имя входного файла: gcd.in  
Имя выходного файла: gcd.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

*Наибольшим общим делителем* двух натуральных чисел  $a$  и  $b$  называется наибольшее натуральное число  $d$  такое, что  $a$  и  $b$  делятся на  $d$  нацело без остатка.

Найдите наибольший общий делитель двух данных чисел.

#### Формат входных данных

В первой строке входного файла заданы через пробел два натуральных числа  $a$  и  $b$  ( $1 \leq a, b \leq 100$ ).

#### Формат выходных данных

В первой строке выходного файла выведите одно число — наибольший общий делитель  $a$  и  $b$ .

#### Примеры

gcd.in	gcd.out
2 3	1
4 2	2

## 4 Задача D. Число Фибоначчи

Имя входного файла: `fib.in`  
Имя выходного файла: `fib.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 Мебибайт

Числа Фибоначчи  $F_0, F_1, F_2, \dots, F_n$  определяются следующим образом:  $F_0 = F_1 = 1$ , а для любого  $n > 1$  выполнено равенство  $F_n = F_{n-1} + F_{n-2}$ .

По заданному числу  $n$  выведите число Фибоначчи  $F_n$ .

### Формат входных данных

В первой строке входного файла задано единственное число  $n$  ( $0 \leq n \leq 45$ ).

### Формат выходных данных

Выведите число  $F_n$  в первой строке выходного файла.

### Примеры

<code>fib.in</code>	<code>fib.out</code>
1	1
2	2
3	3
4	5
5	8
6	13

## 5 Задача Е. Два числа

Имя входного файла: `numbers.in`  
Имя выходного файла: `numbers.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Даны два целых числа  $A$  и  $B$  ( $1 \leq A, B \leq 100$ ). Найдите два таких целых числа  $X$  и  $Y$ , что выполнено равенство  $AX + BY = 1$ .

### Формат входных данных

Во входном файле записаны два числа  $A$  и  $B$ , разделённые пробелом.

### Формат выходных данных

В выходной файл выведите два числа  $X$  и  $Y$ , разделённые пробелом. Требуется, чтобы выполнялись неравенства  $|X| \leq 10\,000$ ,  $|Y| \leq 10\,000$ . Если правильных ответов несколько, разрешается вывести любой из них. Если же таких чисел не существует, выведите вместо них два нуля.

### Примеры

<code>numbers.in</code>	<code>numbers.out</code>
2 3	2 -1
4 6	0 0
100 51	-5075 9951

## 6 Задача F. Самая большая буква

Имя входного файла: `greatest.in`  
Имя выходного файла: `greatest.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дана строка. Найдите в ней букву, алфавитный номер которой — самый большой среди букв этой строки.

### Формат входных данных

В первой строке входного файла задана строка длины от 1 до 100 символов, включительно. Строка состоит из строчных букв латинского алфавита.

### Формат выходных данных

В первой строке выходного файла выведите одну букву — ту букву строки, алфавитный номер которой максимален.

### Примеры

<code>greatest.in</code>	<code>greatest.out</code>
<code>aab</code>	<code>b</code>
<code>dabux</code>	<code>y</code>

## Сложные задачи

### 7 Задача G. Сумма значений функции

Имя входного файла: `valsum.in`  
Имя выходного файла: `valsum.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Найдите сумму значений функции

$$f(x) = x + \frac{1}{x}$$

в нескольких целых точках.

#### Формат входных данных

В первой строке входного файла задано целое число  $n$  — количество точек ( $1 \leq n \leq 50$ ). В следующей строке заданы  $n$  целых чисел  $x_1, x_2, \dots, x_n$  через пробел — точки, значения функции в которых нужно просуммировать ( $0 < |x_i| \leq 10^9$ ).

#### Формат выходных данных

Выведите в выходной файл одно число — сумму значений функции  $f(x)$  в заданных точках. Ответ считается правильным, если абсолютная или относительная погрешность не превышает  $10^{-9}$ .

#### Примеры

<code>valsum.in</code>	<code>valsum.out</code>
3 1 2 3	7.833333333333333
2 1 -1	0



## 8 Задача Н. Очень большие числа

Имя входного файла: `extralarge.in`  
Имя выходного файла: `extralarge.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

А ещё в джаве есть  
Большие числа!

---

народное творчество программистов

Дано два числа. Два **БОЛЬШИХ** числа.

К счастью, с ними не требуется производить сложных арифметических операций. Требуется лишь сравнить их.

### Формат входных данных

Во входном файле записаны две строки, содержащие по одному неотрицательному целому числу, не превышающему  $10^9$  и несколько знаков «!» после него. Каждый знак «!» соответствует операции «факториал». Длины строк не превышают 50.

### Формат выходных данных

Выведите одну из трёх строк:

- « $x > y$ », если первое выражение больше второго;
- « $x < y$ », если первое выражение меньше второго;
- « $x = y$ », если первое выражение равно второму.

### Примеры

<code>extralarge.in</code>	<code>extralarge.out</code>
0! 1	$x = y$
9! 999999999	$x < y$
456!!! 123!!!!!!	$x < y$
5! 120	$x = y$

## 9 Задача I. Лишние пробелы

Имя входного файла: `spaces.in`  
Имя выходного файла: `spaces.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дана строка. Напишите программу, которая удалит из этой строки все лишние пробелы. Пробел будем считать лишним, если:

1. он находится в самом начале строки, до самого первого слова;
2. он находится в конце строки, после самого последнего слова;
3. несколько пробелов расположены между двумя словами (проще говоря, если слова разделены более чем одним пробелом, тогда все пробелы кроме одного — лишние).

### Формат входных данных

Во входном файле записана строка, длина которой не превышает 200 символов. Строка содержит только маленькие латинские буквы и пробелы.

### Формат выходных данных

Выведите в выходной файл эту строку без лишних пробелов.

### Примеры

<code>spaces.in</code>	<code>spaces.out</code>
<code>first test</code>	<code>first test</code>

## 10 Задача J. Коробки

Имя входного файла: `boxes.in`  
Имя выходного файла: `boxes.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Перед Васей cтоят коробки с конфетами. Васе не очень нравится, что не во всех коробках число конфет совпадает. Поэтому он решил съесть несколько конфет.

Вася считает минимальное ( $c_{min}$ ) и максимальное ( $c_{max}$ ) число конфет в коробке. Если  $c_{min} = c_{max}$ , он завершает поедание, иначе он съедает из коробки с  $c_{max}$  конфетами  $c_{min}$  конфет и повторяет процесс.

Определите, сколько конфет будет в каждой из коробок, когда процесс завершится.

### Формат входных данных

В первой строке записано число  $n$  — число коробок ( $1 \leq n \leq 50$ ). Во второй строке записано  $n$  целых чисел от 1 до 1 000 — количества конфет в коробках.

### Формат выходных данных

Выведите единственное целое число — количество конфет в коробках по окончании процесса.

### Примеры

<code>boxes.in</code>	<code>boxes.out</code>
1 7	7
3 7 21 14	7
5 3 4 5 6 7	1
12 366 549 915 183 549 549 183 366 915 549 915 366	183

## 11 Задача К. Тестирующая система

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

Юный программист Саша написал свою первую тестирующую систему. Он так обрадовался тому, что она скомпилировалась, что решил пригласить школьных друзей на свой собственный конкурс.

Но в конце тура выяснилось, что система не умеет сортировать команды в таблице результатов. Помогите Саше реализовать эту сортировку.

Команды упорядочиваются по правилам АСМ:

- по количеству решённых задач в порядке убывания;
- при равенстве количества решённых задач — по штрафному времени в порядке возрастания;
- при прочих равных — по номеру команды в порядке возрастания.

### Формат входных данных

Первая строка содержит натуральное число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество команд, участвующих в конкурсе. В  $i$ -й из следующих  $n$  строк записано количество решённых задач  $S$  ( $1 \leq S \leq 100$ ) и штрафное время  $T$  ( $1 \leq T \leq 100\,000$ ) команды с номером  $i$ .

### Формат выходных данных

В выходной файл выведите  $n$  чисел — номера команд в отсортированном порядке.

### Пример

<code>ejudge.in</code>	<code>ejudge.out</code>
5	5 2 1 3 4
3 50	
5 720	
1 7	
0 0	
8 500	