

## Задача 5. Три сына

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

Во владениях короля Флатландии находится прямая дорога длиной  $n$  километров, по одну сторону от которой расположен огромный лесной массив. Король Флатландии проникся идеями защиты природы и решил превратить свой лесной массив в заповедник. Но сыновья стали сопротивляться: ведь им хотелось получить эти земли в наследство.

У короля три сына: младший, средний и старший. Король решил, что в заповедник не войдут участки лесного массива, которые он оставит сыновьям в наследство. При составлении завещания король хочет, чтобы для участков выполнялись следующие условия:

- каждый участок должен иметь форму квадрата, длина стороны которого выражается целым положительным числом. Одна из сторон каждого квадрата должна лежать на дороге. Пусть участки имеют размеры  $a \times a$ ,  $b \times b$  и  $c \times c$ ;
- стороны квадратов должны полностью покрывать дорогу: величина  $a + b + c$  должна быть равна  $n$ ;
- участок младшего сына должен быть строго меньше участка среднего сына, а участок среднего сына должен, в свою очередь, быть строго меньше участка старшего сына, то есть должно выполняться неравенство  $a < b < c$ ;
- суммарная площадь участков  $a^2 + b^2 + c^2$  должна быть минимальна.

Требуется написать программу, которая по заданной длине дороги определяет размеры участков, которые следует выделить сыновьям короля.

### Формат входного файла

Входной файл содержит одно целое число  $n$  ( $6 \leq n \leq 10^9$ ).

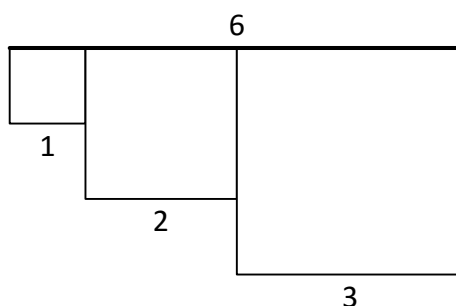
### Формат выходного файла

Выходной файл должен содержать три целых положительных числа, разделенных пробелами:  $a$ ,  $b$  и  $c$  – длины сторон участков, которые следует выделить младшему, среднему и старшему сыну, соответственно. Если оптимальных решений несколько, разрешается вывести любое.

### Пример входных и выходных файлов

<code>division.in</code>	<code>division.out</code>
6	1 2 3

### Пояснение к примеру



### **Описание подзадач и системы оценивания**

В этой задаче четыре подзадачи. Баллы за подзадачу начисляются только в случае, если все тесты для данной подзадачи пройдены.

#### **Подзадача 1 (25 баллов)**

$$n \leq 50.$$

#### **Подзадача 2 (25 баллов)**

$$n \leq 2000.$$

#### **Подзадача 3 (25 баллов)**

$$n \leq 40\,000.$$

#### **Подзадача 4 (25 баллов)**

$$n \leq 10^9.$$

### **Получение информации о результатах окончательной проверки**

По запросу сообщается результат окончательной проверки на каждом тесте.

## Задача 6. Гипершашки

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

Андрей работает судьей на чемпионате по гипершашкам. В каждой игре в гипершашки участвует три игрока. По ходу игры каждый из игроков набирает некоторое положительное целое число баллов. Если после окончания игры первый игрок набрал  $a$  баллов, второй —  $b$ , а третий  $c$ , то говорят, что игра закончилась со счетом  $a:b:c$ .

Андрей знает, что правила игры гипершашек устроены таким образом, что в результате игры баллы любых двух игроков различаются не более чем в  $k$  раз.

После матча Андрей показывает его результат, размещая три карточки с очками игроков на специальном табло. Для этого у него есть набор из  $n$  карточек, на которых написаны числа  $x_1, x_2, \dots, x_n$ . Чтобы выяснить, насколько он готов к чемпионату, Андрей хочет понять, сколько различных вариантов счета он сможет показать на табло, используя имеющиеся карточки.

Требуется написать программу, которая по числу  $k$  и значениям чисел на карточках, которые имеются у Андрея, определяет количество различных вариантов счета, которые Андрей может показать на табло.

### Формат входного файла

Первая строка входного файла содержит два целых числа:  $n$  и  $k$  ( $3 \leq n \leq 100\,000$ ,  $1 \leq k \leq 10^9$ ).

Вторая строка входного файла содержит  $n$  целых чисел  $x_1, x_2, \dots, x_n$  ( $1 \leq x_i \leq 10^9$ ).

### Формат выходного файла

Выходной файл должен содержать одно целое число — искомое количество различных вариантов счета.

### Пример входных и выходных файлов

<code>game.in</code>	<code>game.out</code>
5 2 1 1 2 2 3	9

### Пояснение к примеру

В приведенном примере Андрей сможет показать следующие варианты счета: 1:1:2, 1:2:1, 2:1:1, 1:2:2, 2:1:2, 2:2:1, 2:2:3, 2:3:2, 3:2:2. Другие тройки чисел, которые можно составить с использованием имеющихся карточек, не удовлетворяют заданному условию, что баллы любых двух игроков различаются не более чем в  $k = 2$  раза.

### **Описание подзадач и системы оценивания**

В этой задаче четыре подзадачи. Баллы за подзадачу начисляются только в случае, если все тесты для данной подзадачи пройдены.

**Внимание!** Тест из примера не подходит под ограничения для подзадач 1 и 3, но решение принимается на проверку только в том случае, если оно выводит правильный ответ на тесте из примера. Решение должно выводить правильный ответ на тест, даже если оно рассчитано на решение только каких-либо из подзадач 1 и 3.

#### **Подзадача 1 (15 баллов)**

$$3 \leq n \leq 100\,000, k = 1, 1 \leq x_i \leq 100\,000$$

#### **Подзадача 2 (23 балла)**

$$3 \leq n \leq 100, 1 \leq k \leq 100, 1 \leq x \leq 100$$

#### **Подзадача 3 (30 баллов)**

$$3 \leq n \leq 100\,000, 1 \leq k \leq 10^9, 1 \leq x_i \leq 10^9, \text{ все } x_i \text{ различны.}$$

#### **Подзадача 4 (32 балла)**

$$3 \leq n \leq 100\,000, 1 \leq k \leq 10^9, 1 \leq x \leq 10^9$$

### **Получение информации о результатах окончательной проверки**

По запросу сообщается результат окончательной проверки на каждом тесте.

## Задача 7. Интересные числа

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

Софья считает число *интересным*, если его цифры идут в неубывающем порядке. Например, числа 123, 1111 или 888999 – интересные.

Софья заинтересовалась, сколько существует интересных положительных чисел, лежащих в диапазоне от  $L$  до  $R$  включительно. Это число может оказаться довольно большим для больших  $L$  и  $R$ , поэтому Софья хочет найти остаток от деления этого числа на  $10^9 + 7$ .

Требуется написать программу, которая по заданным  $L$  и  $R$  определяет количество интересных чисел, лежащих в диапазоне от  $L$  до  $R$  включительно, и выводит остаток от деления этого числа на  $10^9 + 7$ .

### Формат входного файла

Входной файл содержит две строки. Первая строка содержит число  $L$ , вторая строка содержит число  $R$  ( $1 \leq L \leq R \leq 10^{100}$ ).

### Формат выходного файла

Выходной файл должен одно целое число – остаток от деления количества интересных чисел, лежащих в диапазоне от  $L$  до  $R$  включительно, на  $10^9 + 7$ .

### Примеры входных и выходных файлов

<code>numbers.in</code>	<code>numbers.out</code>
1 100	54

### Описание подзадач и системы оценивания

#### Подзадача 1 (21 балл)

$$L = 1, R \leq 1000$$

Баллы за подзадачу начисляются только в случае, если все тесты подзадачи пройдены.

#### Подзадача 2 (до 22 баллов)

$$1 \leq L \leq R \leq 10^{18}$$

В этой подзадаче 11 тестов, каждый тест оценивается в 2 балла. Баллы за каждый тест начисляются независимо.

#### Подзадача 3 (до 24 баллов)

$$L = 1, R = 10^k \text{ для некоторого целого } k, 2 \leq k \leq 100.$$

В этой подзадаче 8 тестов, каждый тест оценивается в 3 балла. Баллы за каждый тест начисляются независимо.

#### Подзадача 4 (до 33 баллов)

$$1 \leq L \leq R \leq 10^{100}$$

В этой подзадаче 11 тестов, каждый тест оценивается в 3 балла. Баллы за каждый тест начисляются независимо.

### Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

## Задача 8. Гармоничная последовательность

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

Цикл лекций в университете Флатландии посвящен изучению последовательностей.

Профессор называет последовательность целых чисел  $a_1, a_2, \dots, a_n$  гармоничной, если каждое число, кроме  $a_1$  и  $a_n$ , равно сумме соседних:  $a_2 = a_1 + a_3$ ,  $a_3 = a_2 + a_4$ , ...,  $a_{n-1} = a_{n-2} + a_n$ . Например, последовательность  $[1, 2, 1, -1]$  является гармоничной, поскольку  $2 = 1 + 1$ , и  $1 = 2 + (-1)$ .

Рассмотрим последовательности равной длины:  $A = [a_1, a_2, \dots, a_n]$  и  $B = [b_1, b_2, \dots, b_n]$ . Расстоянием между этими последовательностями будем называть величину  $d(A, B) = |a_1 - b_1| + |a_2 - b_2| + \dots + |a_n - b_n|$ . Например,  $d([1, 2, 1, -1], [1, 2, 0, 0]) = |1 - 1| + |2 - 2| + |1 - 0| + |-1 - 0| = 0 + 0 + 1 + 1 = 2$ .

В конце лекции профессор написал на доске последовательность из  $n$  целых чисел  $B = [b_1, b_2, \dots, b_n]$  и попросил студентов в качестве домашнего задания найти гармоничную последовательность  $A = [a_1, a_2, \dots, a_n]$ , такую, что  $d(A, B)$  минимально. Чтобы облегчить себе проверку, профессор просит написать в качестве ответа только искомое минимальное расстояние  $d(A, B)$ .

Требуется написать программу, которая по заданной последовательности  $B$  определяет, на каком минимальном расстоянии от последовательности  $B$  найдется гармоничная последовательность  $A$ .

### Формат входного файла

Первая строка входного файла содержит целое число  $n$  – количество элементов в последовательности ( $3 \leq n \leq 300\,000$ ).

Вторая строка содержит  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $-10^9 \leq b_i \leq 10^9$ ).

### Формат выходного файла

Выходной файл должна содержать одно целое число: минимальное возможное расстояние от последовательности во входном файле до гармоничной последовательности.

### Примеры входных и выходных файлов

<code>sequence.in</code>	<code>sequence.out</code>
4 1 2 0 0	2

### Пояснение к примеру

В приведенном примере оптимальной является, например, гармоничная последовательность  $[1, 2, 1, -1]$ .

### **Описание подзадач и системы оценивания**

В этой задаче пять подзадач. Баллы за подзадачу начисляются только в случае, если все тесты для данной подзадачи пройдены.

**Внимание!** Тест из примера не подходит под ограничения для подзадачи 1, но решение принимается на проверку только в том случае, если оно выводит правильный ответ на тесте из примера. Решение должно выводить правильный ответ на тест, даже если оно рассчитано на решение только подзадачи 1.

#### **Подзадача 1 (14 баллов)**

$$n = 3, -10 \leq b_i \leq 10$$

#### **Подзадача 2 (14 баллов)**

$$3 \leq n \leq 500, -100 \leq b_i \leq 100$$

#### **Подзадача 3 (16 баллов)**

$$3 \leq n \leq 100\,000, -100 \leq b_i \leq 100$$

#### **Подзадача 4 (16 баллов)**

$$3 \leq n \leq 1000, -10^9 \leq b_i \leq 10^9$$

#### **Подзадача 5 (40 баллов)**

$$3 \leq n \leq 300\,000, -10^9 \leq b_i \leq 10^9$$

### **Получение информации о результатах окончательной проверки**

По запросу сообщается баллы за каждую подзадачу.