

# Всероссийская олимпиада школьников по информатике

Вологодская область, 2024-2025 учебный год

## II (муниципальный) этап

7 – 8 классы

### Задача 1. Города без торговых связей (100 баллов)

В некотором царстве имеются 8 городов, пронумерованных от 1 до 8. В каждом городе имеется ровно один односторонний магический портал, позволяющий перемещать торговые грузы в некоторый другой город.

Вам дан список всех порталов:  $1 \rightarrow 4$ ,  $2 \rightarrow 4$ ,  $3 \rightarrow 1$ ,  $4 \rightarrow 5$ ,  $5 \rightarrow 2$ ,  $6 \rightarrow 4$ ,  $7 \rightarrow 5$ ,  $8 \rightarrow 1$ .

Царь хочет узнать, много ли имеется таких пар городов, между которыми совсем нет связей — то есть нельзя ни из первого города добраться во второй, ни из второго в первый.

Запишите в ответ все такие пары. Каждая пара чисел пишется в отдельной строке, числа в парах разделяются пробелом. Пары и числа в них могут записываться в любом порядке.

Например, если бы у нас было 5 городов и три портала:  $1 \rightarrow 2$ ,  $2 \rightarrow 3$  и  $4 \rightarrow 5$ , тогда возможный правильный ответ был бы таким:

1 4  
1 5  
2 4  
2 5  
3 4  
3 5

**Примечание.** Чем больше правильных пар вы укажете, тем больше баллов получите. За неправильные пары и за дубликаты баллы снижаются. У вас будет только три попытки сдать решение этой задачи в систему, в зачёт идёт лучшее решение из отправленных.

### Задача 2. Сортировка (100 баллов)

На доске написана следующая последовательность чисел:

1 3 4 9 10 2 6 7 8 5

За один шаг можно взять два её любых непересекающихся отрезка одинаковой длины и поменять их местами. Требуется отсортировать последовательность по возрастанию, сделав как можно меньше шагов.

Для каждого шага в отдельной строке напишите через пробел три числа: элементы, с которых начинаются заменяемые отрезки, и длина отрезка. Обратите внимание — пишутся значения элементов, а не их позиции.

Например, если бы нам была дана последовательность 3 4 5 2 1, то возможный верный ответ был бы такой:

3 5 2  
5 1 1

**Примечание.** Чем меньше шагов будет сделано, тем больше баллов получит решение.

У вас будет только три попытки сдать решение этой задачи в систему, в зачёт идёт лучшее решение из отправленных.

### Задача 3. Снова ремонт (100 баллов)

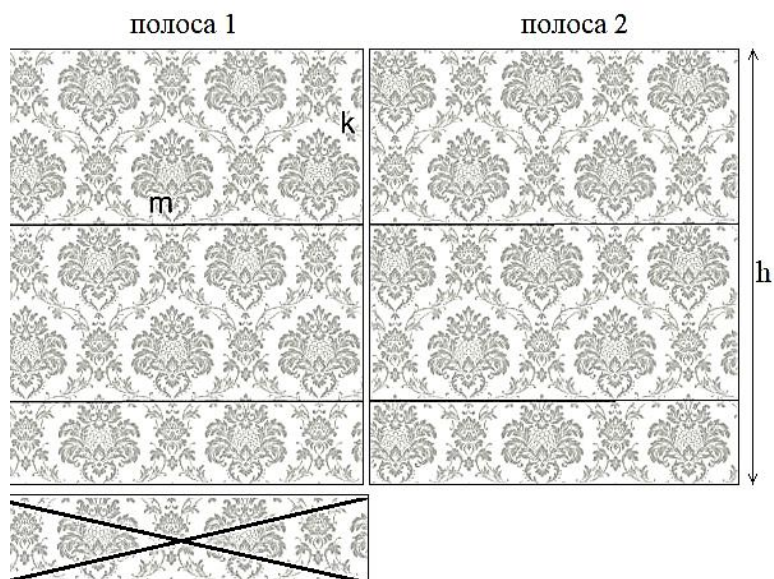
ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод / вывод: стандартный

Маша хочет оклеить обоями стену без окон и дверей длиной  $a$  и высотой  $h$  сантиметров. Маша уже выбрала красивые обои с рисунком, и теперь ей нужно сосчитать, сколько рулонов купить.

Каждый рулон имеет ширину  $m$  сантиметров и длину  $s$  сантиметров. На обоях имеется рисунок, который повторяется вдоль рулона через каждые  $k$  сантиметров.

Процесс оклейки происходит следующим образом. От рулона отрезается полоса длиной  $h$ , она будет приклеена на стену вертикально от пола до потолка. Затем отрезается следующая полоса для приклеивания рядом, и так далее. Однако, не всё так просто! Чтобы получилось красиво, нужно, чтобы рисунок на следующей полосе совместился с рисунком на предыдущей. А чтобы этого добиться, перед отрезанием следующей полосы может понадобиться вначале отрезать лишний кусок рулона (его придётся выбросить).

На рисунке показан пример. От длинного рулона отрезаны две полосы для наклеивания. При этом после отрезания первой полосы пришлось отрезать от рулона ещё кусочек (он показан перечёркнутым крест-накрест), чтобы рисунок на следующей полосе начинался точно так же, как и на предыдущей.



Заметим, что горизонтальные стыки на стенах плохо выглядят, поэтому длина приклеиваемой полосы должна всегда строго равняться высоте комнаты  $h$ . Если последняя приклеиваемая полоса окажется шире, чем оставшаяся длина комнаты, то её можно разрезать вдоль. Также заметим, что все рулоны в магазине абсолютно одинаковые.

Определите, сколько рулонов обоев надо купить Маше, чтобы ей хватило оклеить стену и при этом не взять лишние рулоны.

**Входные данные.** В пяти строках входных данных записаны пять целых чисел:  $a$  – длина стены,  $h$  – высота стены,  $k$  – длина повторения рисунка,  $m$  – ширина рулона обоев,  $s$  – длина рулона. Ограничения:  $1 \leq a, m, s \leq 2 \cdot 10^9$ ,  $h \leq s$ ,  $k \leq s$ .

**Выходные данные.** Выведите одно целое число – минимальное количество рулонов обоев, необходимое для оклейки стены.

### Примеры

**входные данные**

9  
3  
3  
3  
20

**выходные данные**

1

**входные данные**

5  
4  
1  
2  
8

**выходные данные**

2

### Система оценивания.

Решения, правильно работающие при дополнительном ограничении  $a, m, s \leq 1000$ , будут набирать не менее 50 баллов.

#### Задача 4. Согласование заявок (100 баллов)

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод / вывод: стандартный

Владимир Борисович работает руководителем в одной очень крупной Компании. В его служебные обязанности в числе прочего входит рассмотрение и согласование заявок, которые отправляют сотрудники через специальную информационную систему. Владимир Борисович согласовывает заявки разных типов: всего имеется  $n$  типов заявок. Он зашёл в приложение для согласования заявок и увидел, что там имеется  $a_1$  заявок первого типа,  $a_2$  второго и т.д.

Процесс согласования выглядит следующим образом: заявки отмечаются галочкой и нажимается кнопка «Согласовать». Кнопка работает, только когда все отмеченные заявки имеют один и тот же тип, то есть нельзя одновременно согласовать несколько заявок разных типов. Согласованные заявки исчезают из списка заявок. Нажатием на кнопку-галочку возле заявки можно не только отметить её, но и снять отметку. Есть ещё кнопка «Отметить все», которая выставляет галочки напротив всех заявок в отмеченное положение.

За какое наименьшее количество нажатий кнопок в приложении Владимир Борисович может согласовать все заявки?

**Входные данные.** В первой строке входных данных записано число  $n$  — количество типов заявок ( $1 \leq n \leq 10^5$ ). В следующих  $n$  строках записано по одному числу:  $a_1, a_2, a_3$  и так далее — количество поступивших заявок каждого типа ( $1 \leq a_i \leq 10^4$  для всех  $i$ ).

**Выходные данные.** В результате работы программы требуется вывести единственное число — минимальное возможное количество нажатий кнопок для согласования всех заявок.

#### Примеры

##### входные данные

2  
3  
2

##### выходные данные

5

##### входные данные

3  
1  
1  
1

##### выходные данные

6

**Пояснение к примеру.** В первом примере выгоднее всего отметить сначала две заявки второго типа и нажать кнопку «Согласовать», потратив на это суммарно три нажатия на кнопки, после этого нажать кнопку «Отметить все» и затем кнопку «Согласовать», потратив ещё два нажатия, и таким образом согласовать все заявки, используя пять нажатий кнопок. Нетрудно проверить, что любые другие способы согласования требуют большего количества нажатий.

#### **Система оценивания.**

Решения, правильно работающие при дополнительном ограничении  $n \leq 1000$ , будут набирать не менее 70 баллов.

### **Задача 5. Треугольники (100 баллов)]**

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод / вывод: стандартный

Всё, что требуется в этой задаче – найти количество треугольников, у которых длины всех сторон – целые числа в диапазоне от  $P$  до  $Q$  включительно. Например, для  $P=1$ ,  $Q=2$  ответом будет 3 – это треугольники с длинами сторон  $(1, 1, 1)$ ,  $(1, 2, 2)$  и  $(2, 2, 2)$ .

Примечание: тройку  $(1, 2, 2)$  можно было бы записать как  $(2, 1, 2)$  или  $(2, 2, 1)$  – это всё один и тот же треугольник.

**Входные данные.** Вводятся два целых числа  $P$  и  $Q$ , каждое число в отдельной строке ( $1 \leq P \leq Q \leq 2 \cdot 10^5$ ).

**Выходные данные.** Выведите одно целое число – искомое количество треугольников.

#### **Пример**

**входные данные**

1  
2

**выходные данные**

3

**Система оценивания.** Решения, правильно работающие при  $Q \leq 100$ , будут набирать не менее 30 баллов. Решения, правильно работающие при  $Q \leq 2000$ , будут набирать не менее 60 баллов.

**Примечание.** Обратите внимание, что ответ может превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битный целочисленный тип данных (тип `int64` в языке Pascal, тип `long long` в C++, тип `long` в Java и C#). В языке Python ничего дополнительно делать не требуется.