

## Задача 1. Границы массива

Источник:	базовая
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

При работе с массивами многие языки (такие как Java или C) требуют от пользователя описывать размер массива, задающий количество элементов в нем. Если массив описан размерности  $k$ , то он может содержать не более  $k$  элементов, а нумерация индексов элементов изменяется от нуля до  $(k - 1)$ . Так, если массив описан размерности 10, то допустимыми индексами являются числа от 0 до 9.

Ваша задача – написать программу, которая определяет, является ли допустимым индекс для заданного размера массива.

### Формат входных данных

Во входном файле записано через пробел два целых числа  $k$  и  $x$  ( $1 \leq k \leq 10^6$ ,  $-10^6 \leq x \leq 10^6$ ).

Первое целое  $k$  представляет описанный размер массива.

Второе целое  $x$  задает индекс в этом массиве.

### Формат выходных данных

В выходной файл необходимо вывести одну из двух строк.

Если заданный индекс  $x$  является допустимым для данного массива, то нужно вывести строку `legal`.

В противном случае необходимо вывести фразу `out of bounds`.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
10 9	<code>legal</code>
10 10	<code>out of bounds</code>

## Задача 2. Произведение трех чисел

Источник: базовая  
Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: разумное

Вам дано три вещественных числа  $A$ ,  $B$  и  $C$ . Найдите их произведение.

### Формат входных данных

Первая строка входного файла содержит вещественное число  $A$ , вторая строка – число  $B$ , а третья –  $C$  ( $-100.0 \leq A, B, C \leq 100.0$ ).

### Формат выходных данных

В выходной файл необходимо вывести произведение данных чисел.

### Пример

<code>input.txt</code>	<code>output.txt</code>
1.0 1.25 2.5	3.125000

## Задача 3. Минимальное из трех

Источник: базовая  
Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: разумное

Вам дано три целых числа. Найдите среди них минимальное.

### Формат входных данных

Входной файл состоит из трех строк, каждое из которых содержит целое число, по модулю не превосходящее 100.

### Формат выходных данных

В выходной файл необходимо вывести минимальное из данных чисел.

### Пример

<code>input.txt</code>	<code>output.txt</code>
1 2 3	1

## Задача 4. Деление нацело

Источник: основная  
Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: разумное

В файле записано два целых числа  $A$  и  $B$ . Оба числа по модулю не превышают 10 000, число  $B$  положительное.

Требуется найти и вывести четыре целых числа:

1. частное  $A/B$ , округлённое вниз;
2. частное  $A/B$ , округлённое вверх;
3. частное  $A/B$ , округлённое в сторону нуля;
4. остаток  $R$  от деления  $A$  на  $B$ , лежащий в диапазоне  $0 \leq R < B$ .

### Пример

<code>input.txt</code>	<code>output.txt</code>
7 3	2 3 2 1

### Пояснение к примеру

В примере 7 при делении на 3 даёт частное 2 и остаток 1. Если же округлять вверх, то частное  $7/3$  равно 3. Округление в сторону нуля в данном примере работает так же, как округление вниз.

## Задача 5. Треугольник

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

По координатам трех вершин некоторого треугольника найти его площадь и периметр.

### Формат входных данных

Входной файл состоит из трех строк. Каждая строка содержит по два целых числа, по модулю не превосходящих 10000 – координаты соответствующей вершины треугольника. Гарантируется, что все три вершины не лежат на одной прямой.

### Формат выходных данных

В первую строку выходного файла нужно вывести периметр заданного треугольника, а во вторую – его площадь. Вещественные числа выводить с точностью до трех знаков.

### Пример

<code>input.txt</code>	<code>output.txt</code>
0 0	12.000
0 3	6.000
4 0	

### Комментарий

Рекомендуется использовать тип `double` для хранения вещественных чисел, а выводить их с помощью формата `"%0.3lf"`, например:

```
double answer = 0.123456789;  
printf("%0.3lf", answer);
```

## Задача 6. Площадь поверхности

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

В файле записано три целых числа  $x$ ,  $y$  и  $z$ , по модулю не превышающих 100. Требуется определить площадь поверхности прямоугольного параллелепипеда с размерами (т.е. длинами сторон)  $x$ ,  $y$  и  $z$ .

В данной задаче от вас также требуется обработать ошибочные входные данные. Считается, что размер задан ошибочно, если он не является положительным (отрицательный или нулевой).

Если все размеры  $x$ ,  $y$  и  $z$  положительные, нужно вывести одно целое число: искомую площадь поверхности. В противном случае нужно вывести со знаком минус номер ошибочного размера. Размер  $x$  имеет номер 1, размер  $y$  имеет номер 2, а размер  $z$  имеет номер 3. Если ошибочных размеров несколько, выберите размер с наименьшим номером.

### Пример

<code>input.txt</code>	<code>output.txt</code>
3 4 5	94
-1 -1 -1	-1

### Пояснение к примеру

Во втором примере все три размера ошибочные. Из трёх вариантов -1, -2, и -3 нужно вывести -1, т.к. размер  $x$  имеет наименьший номер 1.

## Задача 7. Часы

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Часы показывают время в 24-часовом формате. Сначала записывается целое число  $H$  — количество полных часов, прошедших с полуночи ( $0 \leq H \leq 23$ ), затем  $M$  — количество минут, прошедших с начала часа ( $0 \leq M \leq 59$ ), и наконец  $S$  — количество секунд, прошедших в текущей минуте ( $0 \leq S \leq 59$ ).

Нужно, зная текущие показания часов, определить, что они будут показывать через  $K$  секунд ( $1 \leq K \leq 1\,000\,000$ ).

### Формат входных данных

В файле записано четыре целых числа: текущие показания часов  $H$ ,  $M$ ,  $S$ , и количество секунд  $K$ .

### Формат выходных данных

В выходной файл нужно вывести через пробел три целых числа  $H'$ ,  $M'$ ,  $S'$  — показания часов через  $K$  секунд (в том же формате).

### Пример

<code>input.txt</code>	<code>output.txt</code>
12 34 9 57	12 35 6

### Пояснение к примеру

В примере часы показывают 12:34:09, а через 57 секунд они будут показывать 12:35:06. Обратите внимание, что ведущие нули печатать **не** нужно.

## Задача 8. Корни

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

По заданным коэффициентам квадратного уравнения  $ax^2 + bx + c = 0$  надо найти его корни.

### Формат входных данных

В первой строке входного файла записано через пробел три целых числа  $a$ ,  $b$  и  $c$  – коэффициенты квадратного уравнения. Все числа по модулю не превосходят 100.

### Формат выходных данных

Если заданное квадратное уравнение не имеет корней, то в выходной файл нужно вывести слово **NO**.

Если корней бесконечное множество, то вывести **MANY**.

Если уравнение не имеет смысла, то вывести **ERROR**.

Если корень один, то в первую строку выходного файла нужно вывести число 1, а во вторую – значение этого корня с точностью до 5 знаков после десятичной точки.

Если у уравнения два корня, то в первую строку нужно вывести число 2, а во вторую – найденные корни в порядке возрастания их значений с точностью до 5 знаков после десятичной точки.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
0 0 1	ERROR
1 -2 -3	2 -1.00000 3.00000



## Задача 9. Скорая помощь

Источник:	повышенной сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Диспетчер скорой помощи принял вызов. Он спросил адрес, номер квартиры, но забыл узнать номер подъезда и этаж. Эта информация ему необходима для того, чтобы врач быстрее добрался до больного. Диспетчер был опытный. Он вспомнил, что в этот дом уже были вызовы, в которых были указаны и номера квартир, и этажи, и подъезды.

Помогите диспетчеру вычислить этаж и подъезд для квартиры  $N$ , если известно, что квартира с номером  $M$  в этом доме располагается в подъезде  $P$  на этаже  $K$ , а дом имеет  $L$  этажей, и на каждой площадке расположено одинаковое количество квартир.

Гарантируется, что на всех тестах решение существует и единственно.

### Формат входных данных

Первая строка входного файла содержит целое число  $N$  — номер квартиры, откуда поступил вызов ( $0 < N < 500$ ).

Вторая строка состоит из четырех целых чисел  $M, P, K, L$ , записанных через пробел, обозначающих, соответственно, номер квартиры, подъезд и этаж, а также количество этажей в доме ( $P > 1$ ).

### Формат выходных данных

Выходной файл должен состоять из одной строки, в которую через пробел записаны два целых числа — номер подъезда и этаж для квартиры  $N$ .

### Пример

<code>input.txt</code>	<code>output.txt</code>
100 50 2 4 9	3 7

### Пояснение к примеру

В примере количество квартир на этаже равно 4, а 50-ая квартира является второй на четвёртом этаже второго подъезда. Тогда квартира с номером 100 является четвёртой на седьмом этаже третьего подъезда.

## Задача 10. Цифра 0 или 1

Источник: повышенной сложности  
Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: разумное

Дано натуральное  $k$ .

Определить  $k$ -ю цифру в последовательности 110100100010000100000..., в которой выписаны подряд степени 10.

### Формат входных данных

Во входном файле записано число  $k$  ( $1 \leq k \leq 10^8$ ).

### Формат выходных данных

В выходной файл нужно вывести ту цифру, которая стоит на  $k$ -м месте в заданной последовательности.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
3	0
4	1

### Замечания

При решении данной задачи циклы не использовать!