

1. Написать функцию **average**, которая в качестве входного параметра принимает массив целых чисел и возвращает их среднее значение.
2. Написать функцию **min**, которая принимает на вход массив вещественных чисел и возвращает минимальный элемент массива.
3. Написать функцию **firstEven**, которая на вход принимает **переменное число параметров** целого типа и возвращает первый найденный элемент из заданной последовательности, который является четным числом.
4. Написать функцию **byteConverter**, принимающую на вход 2 аргумента. Назначение функции – перевести мегабайты в байты или килобайты. Первый аргумент – целое число в диапазоне от 0 до 10^6 , представляющее количество **мегабайт**. Второй аргумент – строка, в которую может быть записано одно из 2-х значений: “b” или “kb”. В зависимости от значения второго аргумента, функция должна возвращать количество байт, либо килобайт, согласно значению первого аргумента.
5. Написать функцию **isSorted**, которая принимает на вход массив элементов целого типа. Функция должна вернуть значение **true**, если заданный массив является отсортированным по возрастанию, либо **false** – в противном случае.
6. Написать функцию **isPrime**, принимающую на вход целое число и возвращающую **true**, если заданное значение является простым числом, либо **false**, если число составное.
7. Написать функцию **sumDigits**, которая на вход принимает целое положительное число и возвращает сумму цифр этого числа.
8. Написать функцию **showTime**, которая принимает на вход количество секунд (целое положительное число). Функция должна перевести время из секунд в часы, минуты и секунды, и вывести значение в формате “HH:MM:SS” на экран. Например, при вызове функции `showTime(765)`, на экран должна вывести надпись “00:12:45”, т.к. аргумент 765 соответствует 12 минутам и 45 секундам. Обратите внимание, что если один из параметров времени меньше 10, то перед ним необходимо вывести лидирующий 0, чтобы в каждой позиции было записано минимум 2 цифры. Функция не должна возвращать значения, а лишь выводить полученный результат на экран.
9. Написать функцию **replaceNegatives**, которая принимает в качестве аргумента двумерный массив (матрицу) размером `nхm`, состоящий из целых чисел. Функция должна заменить все отрицательные значения в этой матрице на 0 и вернуть

измененную матрицу. *Примечание:* кол-во строк и столбцов в матрице может быть любым и варьироваться в диапазоне от 1 до 20.

10. Написать функцию **sqRoots**, которая должна вывести на экран корни квадратного уравнения. В качестве входных аргументов функция должна принимать 3 вещественных значения a, b и c, которые являются коэффициентами квадратного трехчлена вида

$$ax^2 + bx + c = 0,$$

Если корни для заданных аргументов определить невозможно, то вывести на экран сообщение “**корней нет**”.

P.S. Для вычисления квадратного корня числа вам может понадобиться соответствующая функция, которую вы можете найти в пакете Math. Пример вызова функции просчета квадратного корня:

```
double r = Math.sqrt(16.0); // r = 4.0
```