

## Файлы-функции

Все задания выполнять в ОТДЕЛЬНЫХ файлах-функциях.

**Задание #01.** Создайте файл с именем **average.m**. Скопируйте туда код, представленный ниже. Проверьте работоспособность кода. Измените код, используя конструкцию `try catch` вместо условного оператора. Перепишите код таким образом, чтобы можно было получать среднее значение для матрицы размера `nхm`.

```
function y = average(x)
% AVERAGE Среднее значение элементов вектора.
% AVERAGE(X), где X - вектор.
% Вычисляет среднее значение элементов вектора.
% Если входной аргумент не является вектором, генерируется ошибка.
    [m,n] = size(x);
    if (~(m == 1) | (n == 1) | (m == 1 & n == 1))
        s='Входной массив должен быть вектором'
    end
    y =sum(x)/length(x); % Собственно вычисление
```

Код взят со страницы:

[http://old.exponenta.ru/soft/MATLAB/potemkin/book/matlab/chapter3/3\\_2.asp](http://old.exponenta.ru/soft/MATLAB/potemkin/book/matlab/chapter3/3_2.asp)

**Задание #02.** Представлен код файла-функции **RectangleSP.m**, вычисляющей площадь и периметр прямоугольника по двум входным параметрам – сторонам прямоугольника. Дополните программу анализом, что вводимые положительные, используйте условный оператор, и для генерации сообщения об ошибке - функцию **error**.

```
function [s,p] = RectangleSP(a,b)
s = a*b;
p=2*(a+b);
```

**Задание #03.** Напишите функцию **MySum**, находящую сумму произвольного количества чисел, для этого используйте функцию **varargin**, позволяющую использовать любое количество входных параметров. При вызове функции входные параметры указывать через запятую. Вызовите функцию несколько раз:

```
s=MySum(1,2,3,4,5)
s=MySum(1,2,3,4,5.99)
s=MySum(1,2,3,4, 'u')
s=MySum(1,2,3,4,5,[1 2 3])
s=MySum(1,2,3,4,5,[1;2;3])
s=MySum(1,2,3,4,5,[1 2;3 4])
```

Проанализируйте полученные результаты.

**Задание #04.** Напишите функцию, входными параметрами которой, являются произвольное количество (**varargin**) векторов. Функция должна возвращать максимальный элемент среди элементов всех векторов.

**Задание #05.** Напишите функцию, входными параметрами которой, являются произвольное количество (**varargin**) матриц. Функция должна возвращать максимальный элемент среди элементов всех матриц.

**Задание #06.** Напишите функцию, у которой входной параметр – вектор, а выходных параметров может быть от 1-го до 4-х: значение максимального элемента, значение минимального элемента, индексы всех максимальных элементов, индексы всех минимальных элементов. Для выходных параметров в описании функции используйте (**varargout**). Для обращения к конкретному выходному параметру в теле функции используйте `varargout{k}`, где `k` – это номер выходного параметра. Для успешного решения задачи посмотрите справку по функциям `max` и `find`.

**Задание #07.** Напишите функцию, входными параметрами которой, являются произвольное количество (**varargin**) векторов. Функция должна возвращать среднее значение среди элементов всех векторов.

**Задание #08.** Напишите функцию, входными параметрами которой, являются произвольное количество (**varargin**) матриц. Функция должна возвращать среднее значение среди элементов всех матриц.

**Задание #09.** Напишите функцию, входными параметрами которой, являются произвольное количество (**varargin**) двумерных векторов, представляющих координаты точек на плоскости. Функция по этим точкам строит график.

Заголовок функции должен иметь вид:

```
function testvar(varargin)
```

В теле функции написать цикл по количеству входных параметров. В этом цикле наполнить вектора `x` и `y`, по которым потом будет построен график. Так как каждый входной вектор имеет длину 2, то тело цикла будет состоять из строк:

```
x(i) = varargin{i}(1); % 1-й элемент очередного входного параметра
```

```
y(i) = varargin{i}(2); % 2-й элемент очередного входного параметра
```