Lab. 2 Functions; IF and FOR instructions

Do the exercises below in the Octave IDE. You should only use assignments operations with arithmetic expressions including pre-defined MATLAB functions. Also use scripts to avoid "too much typing".

1. **Classification of a triangle**

   Specify function **type_triangle/3** that takes as arguments three non-negative numbers, interpreted as the sizes of the three sides of a triangle, and returns the type of such rectangle encoded as

   > 0 – not a triangle
   > 1 – scalene triangle
   > 2 – isosceles triangle
   > 3 – equilateral triangle

   Examples:
   ```
   triangle_type([ 6 6 6 ] -> 3
   triangle_type([ 9 2 4 ] -> 0
   triangle_type([ 9 5 5 ] -> 2
   ```

2. **3rd degree Equation**

   Specify function **equation_3/4** that takes as arguments four real numbers, interpreted as the parameters of the equation of the 3rd degree, $ax^3 + bx^2 + cx + d = 0$, and returns its real solutions. <mark>Note: Use the resolvent formula.</mark>

3. **Vector Statistics**

   Specify function v_stat/1 that takes a vector of real numbers as an argument and returns a vector with the **max**, **min**, **mean** and **standard deviation** of the elements of the vector.

   Example:
   ```
   v_stat([ 3, 5, 6, 4, 7] -> [7, 3, 5.0, 1.4142]
   ```

4. **Matrix Statistics**

   Specify function m_stat /1 that takes a matrix of real numbers as an argument and returns a vector with the **max**, **min**, **mean** and **standard deviation** of the elements of the matrix.

   Example:
   ```
   m_stat([ 3, 5, 6; 4, 5, 7] -> [7, 3, 5.0, 1.2910]
   ```

5. **Stability**

   Specify function stable/1 that takes as input a vector of integer numbers and returns as a result a vector indicating how many times an element is less, equal or greater than the previous element.

   Example: `stable([1 2 4 5 3 7 7 2 2 8 6] = [3, 2, 5]`

6. **Averaging rows and columns**

   a) Specify function **c_mean/1** that takes as input matrix of integer numbers and returns as a result a row vector with the same number of columns, each element representing the average of the elements of the matrix in that column

   Example: `c_mean([1 7 2 4; 5 9 0 8] -> [3 8 1 6]`

   b) Specify function r_mean/1 that does the same with the rows.

   Example: `r_mean([1 7 2 4; 5 9 0 8] -> [3.5 5.5]`

7. **Matrix Multiplication**

Specify function **mat_mult/2** that takes as input two matrices with real numbers and returns their product. Note: if the matrices are not *compatible* return an empty array.

Example: Given `A =[4 3 ; 1 2 ; 7 8], B = [0 3 4; 2 1 4]`

       `mat_mult(A,B) -> [6 15 28; 4 5 12, 16 29 60]`

8. **Boolean Matrix Multiplication**

Specify function **bool_mat_mult/2** that takes as input two Boolean matrices and returns their Boolean product (i.e. similar to the numeric case, but replacing multiplication by conjunction and sum by disjunction. Note: if the matrices are not *compatible* return an empty array.

Example: Given `A =[1 0 ; 0 1 ; 1 1], B = [0 1 0; 1 1 0]`

       `bool_mat_mult(A,B) -> [0 1 0; 1 1 0; 1 1 0]`