Exercise 3 - Equations, Algorithms, and Functions

1 AngleTransformer Using Equations

Show that the AngleTransformer model below can be used for computing radians or degrees depending on which variable (rad or deg) a value is given. For example, do this by defining and simulating a model that contains two different instances of AngleTransformer, and use modifier equations to these to specify radians or degrees, respectively.

```
model AngleTransformer
    Real rad;
    Real deg;
protected
    constant Real Pi = Modelica.Constants.pi;
equation
    deg = 180/Pi*rad;
end AngleTransformer;
```

2 Faculty

Write a function `faculty`, using a for-loop, such that faculty(n) = 1*2*3*4*....*n.

```
function ...
```

Write a class that contains a function call to `faculty`.

```
class ...
```

3 Nested for-loop

Write a function, `matrixAddition`, for adding two two-dimensional matrices.

Perform a function call to `matrixAddition` with two matrices. Then simulate the class with the function call and plot the result.

4 Functions and Algorithm Sections

a) Write a function, mySum, which calculates the sum of Real numbers, for a vector of arbitrary size. Hint: you may declare an input vector of arbitrary size as having the type Real[:]; also, it might be convenient to use a for-loop or a while-loop to make the summation.

```
function mySum ...
```

Call sum:

```
mySum(...
```

b) Write a function, average, which calculates the average of Real numbers, in a vector of arbitrary size. The function average should make use of a function call to mySum.

```
function average ...
```

Call average:

```
average(...
```

5 Functions - LargestAverage

Write a class, LargestAverage, that has two vectors and calculates the average of each of them. Then it compares the averages and sets a variable to true if the first vector is larger than the second and otherwise false.