Object Oriented Programming with C++
Labwork 4 – *Operators and Friendship*

A skeleton is provided in the archive file. It allows to start quickly the exercices, and already answer part of the questions. So, use it!

Exercise 1:

Write a class Vector that represents a vector in real space $\mathbb{R}^n$ for any dimension n (so having n real coordinates). Try to think in term of use-case about the different possible constructors (how many, type and default values of their arguments). Complete the following methods:

- The predicate returning true iff two vectors are equals.
- Subtraction of two vectors (binary function, with one argument). Again, take care about possible incompatible dimensions.
- Scalar product by another vector, of same dimension (that is, the sum of the products of their coordinates of same index). How to handle correctly the problem of incompatible (different) dimension?
- By scalar product, that returns a new vector.
- Getters and setters, element by element.

A lot of things are already proposed in the skeleton. In the `Vector.hpp` file, your work starts line 102.

Complete the test program (file `main.cpp`) to test your implementation.

Exercise 2:

Write a class Matrix, which dimensions are $n \times m$, that represents a matrix in $\mathbb{R}^{n\times m}$ (so containing real numbers). Again, you have to find the most useful constructors starting with some use-case. Add the following methods:

- Getters, element by element.
- By matrix product. How to handle correctly the incompatible dimensions problem?
- By vector product. How to handle correctly the incompatible dimensions problem?
- Subtraction of two matrices (returning a new matrix). How to handle correctly the incompatible dimensions problem?
- By scalar product, that returns a new matrix having the elements of the first multiplied by the given scalar. Notice that you have to add this method into the given files. This means also you need to add new test cases into the test program (`main.cpp`).

More precisely, your work start line 118 of file `Matrix.hpp`, and line 15 of file `Matrix.cpp`. Do not forget to copy your solution here from the previous exercise (*i.e.* to copy the `Vector.?pp` files).