SE assignment **Univ. Rennes 1**

Car Immatriculation

We are about to begin our new job at the "European Immatriculation Car Center", and our first task will consist of optimizing the data format for the registration of car plates and owner data. Our boss asks us to initially work on a small-scale example of the database (he provides us with the Java class `carImmatriculation`) and to test some smart new ideas aiming at reducing the number of bytes necessary to store the overall information. It lookes like the Center has almost no more storage capability.

The Data class

One of our new colleagues suggests us to give a look at the public `binMeta` project[1], written in Java, and in particular at its class named `Data`. It seems that this class was already successfully used in other projects developed at the Center.

Storing owner's names

We can remark that the `Data` class provides us with some interesting methods for performing conversions from and to `char` primitive types. Moreover, the `Data` object representing a *letter* of the alphabet only requires 5 bits to be stored, and not the standard 8 bits of the ASCII code. This will help us reducing the total number of necessary bits!

Let's modify therefore the Java class `carImmatriculation` so that owner's names and surnames are stored in one unique `Data` object. Our boss tells us that we do not need to keep the blank character separating the names from the surnames.

But what about the plate numbers?

Our boss is quite satisfied by our results on the compression of owner personal information. However, he admits that he had figured out this solution by himself. What he's really interested to have is a new compression system for the plate numbers. The situation is a little more complicated in this case. In fact, plate numbers do not only contain alphabetic letters, but also decimal digits. How to reduce to the minimum the number of bits necessary to store every plate number?

[1] `https://github.com/mucherino/binMeta`