IUT Littoral **2019 - 2020**
Info 2A **Ana. Num. – C++**

TP 4 - Matrices

Le but de ce TP est de créer une class `Matrice` en `C++` permettant de représenter les matrices à coefficients dans $\mathbb{R}$. La sructure de notre future class est donnée sur ma page.

Le fichier `matrice.hpp` contient les déclarations des fonctions de notre classe ainsi que la défininition des plus basiques (contructeur, recopie, affichage, ...) Normalement vous n'aurez pas besoin de modifier ce fichier. Le fichier `matrice.cpp` recevra la définition des fonctions demandées en exercice. Le prototype de chacune des fonctions à coder est commenté. Vous devrez donc décommenter les prototypes au fur et à mesure de votre progression.

Exercice 1. Parcourire les fichiers `matrice.hpp` et `matrice.cpp` pour prendre connaissance des données de la classe ainsi que les fonctionnalités déjà disponibles.

Exercice 2. Dans le fichier `matrice.cpp`, coder les fonctions suivantes :

1. `bool Matrice::operator==(const Matrice& M)` testant si deux matrices sont égales;
2. `bool Matrice::operator!=(const Matrice& M)` testant si deux matrices sont différentes;
3. `Matrice Matrice::operator+(const Matrice& M)` retournant la somme de deux matrices;
4. `Matrice Matrice::operator-(const Matrice& M)` retournant la dfférence de deux matrices;
5. `Matrice Matrice::operator*(double& a)` retournant le produit d'une matrice par un scalaire `a`;
6. `Matrice Matrice::operator*(const Matrice& M)` retournant le produit de deux matrices;

Exercice 3. Dans le ficheir `matrice.cpp`, coder les fonctions suivantes :

1. `void Matrice::echange_lignes(size_t i,size_t j)` faisant l'opération élémentaire de lignes $L_i \leftrightarrow L_j$;
2. `void Matrice::multiplie_ligne(size_t i,double a)` faisant l'opération élémentaire de lignes $aL_i \rightarrow L_i$,
3. `void Matrice::ajoute_multiple_ligne(size_t i,size_t j,double a)` faisant l'opération élémentaire de ligne $L_i + aL_j \rightarrow L_i$;
4. `void Matrice::Gauss()` retournant la matrice obtenue de la courante après application de la méthode du pivot de Gauss.

Exercice 4. A l'aide du pivot de Gauss, coder les fonctions suivantes :

1. `size_t Matrice::rang()` retournant le rang de la matrice;
2. `double Matrice::determinant()` retournant le determinant de la matrice;
3. `Matrice Matrice::inverse()` retournant l'inverse de la matrice.