

### Exercice sur les classes et pointeurs (pour le 13/11/2014)

*Objectif* : créer une classe `Matrice`, avec partage mémoire (copie superficielle, *shallow copy*), donc avec des copies à faible coût, celles-ci n'engendrant pas d'allocation mémoire.

- Ecrire une classe `Matrice` avec méthodes pour affichage, accès à un élément (en lecture et écriture). On stockera les éléments dans un tableau à une seule dimension `tab`.
- Ajouter un opérateur de multiplications de deux matrices (avec vérification des tailles).
- Ecrire un constructeur, le constructeur par copie et le destructeur. Chacun doit afficher un message unique par `cout`.
- En plus de partager le tableau des pixels, on aura un champ `int* compteur` dans `Matrice` qui sert de compteur au nombre de partages du tableau des coefficients. Ainsi il est alloué et mis à 1 dans le constructeur, incrémenté dans le constructeur par recopie et décrémenté dans le destructeur. Quand il retombe à 0, le destructeur doit tout détruire.
- Dans une boucle, faire des matrices aléatoires, les copier, les multiplier, etc.

Votre programme comportera donc les fichiers `matrice.h`, `matrice.cpp` et `main.cpp`. Envoyer les fichiers source, ainsi que le `CMakeLists.txt`, par mail : `monasse(at)imagine.enpc.fr`