

À rendre par mail avant le 10 mars :
alexandrebenoit@yahoo.fr

Le but de ce DM est de programmer quelques algorithmes sur les matrices.

On complétera le fichier python : **dm2.py** disponible sur le site web. Le fichier devra juste contenir les fonctions demandées.

On testera toutes les fonctions à partir de matrices aléatoires de différentes tailles.

Dans ce DM, une matrice est une liste de listes. La fonction **matrice_aleatoire** permet de créer une matrice contenant des nombres aléatoires entre -99 et 100 . La fonction **matrice_nulle** permet de créer une matrice contenant des 0 .

La fonction **produit_avec_scalaire** retourne le produit d'un nombre avec une matrice. Il est utile de bien comprendre cet algorithme avant de commencer à écrire les fonctions.

Pour chaque fonction, on indiquera en commentaires ou dans un fichier à part les complexités de chaque algorithme.

Exercice 1 :

Écrire la fonction **taille**, qui prend comme argument une matrice, et renvoie le couple de nombres (nb de lignes, nb de colonnes) :

Par exemple pour une matrice M de 5 lignes et 3 colonnes, la fonction **taille(M)** renvoie le couple **(5,3)**.

Dans la suite de ce DM, on utilisera la fonction **taille** pour donner la taille d'une matrice.

Exercice 2 :

- (1) Écrire la fonction **est_carree** qui prend comme argument une matrice et renvoie le booléen **True** si celle-ci est une matrice carrée et **False** sinon.
- (2) Écrire la fonction **est_diagonale** qui prend comme argument une matrice et renvoie le booléen **True** si celle-ci est une matrice diagonale et **False** sinon.
- (3) Écrire la fonction **est_triangulaire_sup** qui prend comme argument une matrice et renvoie le booléen **True** si celle-ci est une matrice triangulaire supérieure et **False** sinon.

Exercice 3 :

Écrire la fonction **addition** qui prend comme arguments deux matrices et renvoie la somme de ces matrices si elles sont de mêmes tailles, sinon elle affiche un message indiquant un problème de taille.

Exercice 4 :

- (1) Écrire la fonction **transpose** qui prend comme argument une matrice de taille (n, q) et renvoie sa matrice transposée de taille (q, n) .
- (2) En déduire la fonction **est_symetrique** qui prend comme argument une matrice et renvoie le booléen **True** si celle-ci est une matrice symétrique et **False** sinon.
- (3) On écrira une fonction équivalente pour les matrices antisymétrique.

Exercice 5 :

- (1) Écrire la fonction **produit** qui prend comme arguments deux matrices de taille respective (n, q) et (q, p) et renvoie la matrice produit de taille (n, p) . Si les tailles ne coïncident pas, la fonction affichera un message d'erreur.
- (2) En déduire la fonction **puissance** qui prend comme arguments une matrice carré M et un nombre entier n positif et renvoie M^n .