# TP Algorithmique – Calcul Matriciel

## Préambule :

Ce TP se déroule sur deux séances (2 fois 1h) et sera ramassé à la fin de la 2ème heure.

Une introduction du sujet a été faite lors de la séance de rattrapage du lundi 7 Janvier 2019.

La première séance ayant été effectuée le 11 Janvier 2019, **le TP sera donc rendu au plus tard le 18 Janvier 2019.**

## Objectif :

*Le but de ce TP est d'implémenter un programme sous Python permettant de calculer des expressions du type* $x_0 \times I_3 + x_1 \times A + x_2 \times A^2 + \cdots + x_n \times A^n$ *pour une matrice A carrée d'ordre 3 et pour des coefficients réels* $x_0$, $x_1$, $x_2$,… $x_n$ *(les coefficients et les éléments de la matrice étant rentrés par l'utilisateur). Nous définirons une matrice :*

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

*comme la liste dont les éléments sont les listes des éléments de chaque ligne :*

$$\left[[a_{1,1}, a_{1,2}, a_{1,3}], [a_{2,1}, a_{2,2}, a_{2,3}], [a_{3,1}, a_{3,2}, a_{3,3}]\right]$$

*Dans tout le TP, toute matrice sera d'ordre 3.*

*On veillera à tester notre programme pour le calcul suivant :*

$$I_3 + 2A + 3A^2 \text{ ;}$$

$$\text{avec } A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}$$

## Questions

1. On souhaite implémenter une fonction **so(A,B)** permettant de calculer la somme de deux matrices.

On propose l'algorithme suivant :

| **Fonction : so**(A,B : listes)<br>**Rôle :** calcule la somme A+B pour A et B matrices carrées d'ordre 3<br>**Variables locales :** i, j (entiers), C(liste) |
|---|
| **Début**<br>$C \leftarrow$ [ [0,0,0], [0,0,0], [0,0,0] ]<br>**Pour i de 0 à 2 Faire**<br>**Pour j de 0 à 2 Faire**<br>$C[i][j] \leftarrow A[i][j] + \ B[i][j]$<br>**FinPour**<br>**FinPour**<br>Retourner (C)<br>**FinFonction** |

**Implémentez en Python la fonction correspondante.**

2. On souhaite implémenter une fonction **produitR**(k,A) qui à un réel $x$ et à une matrice A carrée d'ordre 3, retourne la matrice $kA$.

On propose l'algorithme suivant :

| **Fonction : produitR**(k : réel , A : liste)<br>**Rôle :** calcule le produit $k.A$ pour A matrice carrée d'ordre 3<br>**Variables locales :** i, j (entiers), C(liste) |
|---|
| **Début**<br>$C \leftarrow$ [ [0,0,0], [0,0,0], [0,0,0] ]<br>**Pour i de 0 à 2 Faire**<br>**Pour j de 0 à 2 Faire**<br>$C[i][j] \leftarrow k * A[i][j]$<br>**FinPour**<br>**FinPour**<br>Retourner (C)<br>**FinFonction** |

**Implémentez en Python la fonction correspondante.**

3. On souhaite implémenter une fonction **produit(k,A)** qui à un réel $x$ et à deux matrices A et B carrée d'ordre 3, retourne la matrice $A \times B$.

On propose l'algorithme suivant :

| **Fonction : produit**(A,B : listes)<br>**Rôle :** calcule le produit $AB$ pour A et B matrices carrées d'ordre 3<br>**Variables locales :** i, j, k (entiers), C(liste) |
|---|
| **Début**<br>$C \leftarrow$ [ [0,0,0], [0,0,0], [0,0,0] ]<br>**Pour i de 0 à 2 Faire**<br>**Pour j de 0 à 2 Faire**<br>**Pour k de 0 à 2 Faire**<br>$C[i][j] \leftarrow C[i][j] + A[i][k] * B[k][j]$<br>**FinPour**<br>**FinPour**<br>**FinPour**<br>Retourner (C)<br>**FinFonction** |

**Implémentez en Python la fonction correspondante.**

4. Ecrire une fonction **puissance(A,n)** qui retourne $A^n$, à une matrice A carrée d'ordre 3, et un entier naturel $n$.

5. Ecrire un programme demandant à l'utilisateur un entier naturel $n$ et les coefficients $x_0$, $x_1$, $x_2$,… $x_n$, puis les éléments de la matrice A, carrée d'ordre 3, et affichant la matrice $x_0 \times I_3 + x_1 \times A + x_2 \times A^2 + \cdots + x_n \times A^n$.
Ce programme devra faire appel aux fonctions **so(A ,B), produitR(k, A) et puissance(A,n).**
Implémentez sous Python puis exécuter le programme pour l'exemple suivant :

$$A = [\,[1,2,3], [2,3,4], [3,4,5]\,]$$
$$n = 2$$
$$x_0 = 1,\ x_1 = 2, x_2 = 3$$