

# TD5 - Algorithmique

## Exercice 1 : Matrices

On souhaite écrire différentes fonctions permettant de calculer avec les matrices. Le but de ce TD est de faire écrire des boucles, on vous demande donc de ne pas les « éviter ».

Une matrice est représentée par une liste de listes par exemple la matrice  $M = \begin{pmatrix} -1 & 2 & 3 \\ 2 & 0 & 1.5 \end{pmatrix}$  sera stockée sous la forme suivante :

```
>>> M = [[-1, 2, 3], [2, 0, 1.5]]
>>> M
[[-1, 2, 3], [2, 0, 1.5]]
```

Les listes étant des objets modifiables (on dit aussi mutables), on peut changer directement la valeur d'un coefficient :

```
>>> M[1][1] = -3 # M[i][j] donne accès au coef. en position i, j
>>> M
[[-1, 2, 3], [2, -3, 1.5]]
```

1. Pour vous aider à démarrer, on donne le code de la fonction `MatNulle(n, p)` qui renvoie la matrice nulle de format  $(n, p)$  :

```
def MatNulle(n, p):
    M = []
    for i in range(n):
        L = []
        for j in range(p):
            L.append(0)
        M.append(L)
    return M
```

2. Écrire une fonction `affiche(M)` qui affiche les lignes de la matrice passée en paramètre.
3. Écrire une fonction `format(M)` qui prend en paramètre une matrice `M` et qui renvoie le couple  $(n, p)$  donnant son format.
4. Écrire une fonction `MatIdentite(n)` qui renvoie la matrice identité de taille  $n$ .
5. Écrire une fonction `mult(M, a)` qui prend en paramètre une matrice `M` et qui renvoie le produit de cette matrice par le flottant `a`. La matrice passée en paramètre ne devra pas être modifiée par cette opération.
6. Écrire une fonction `somme(M, N)` qui prend en paramètre deux matrices `M` et `N` de même format et qui renvoie leur somme. La matrice passée en paramètre ne devra pas être modifiée par cette opération.

On commencera par vérifier que les formats sont identiques.

7. Écrire une fonction `prod(M, N)` qui prend en paramètre deux matrices `M` et `N` et qui renvoie le produit de `M` par `N`. Les matrices passées en paramètre ne devront pas être modifiées par cette opération.

On commencera par vérifier que les formats sont compatibles.

## Exercice 2 : Polynômes

---

On souhaite écrire différentes fonctions permettant de calculer avec les polynômes. Un polynôme est représenté par la liste de ces coefficients, par exemple  $f(z) = 3 - \frac{1}{2}z - 2z^2 + 5z^3$  sera représenté par la liste `[3, -0.5, -2, 5]` ainsi le coefficient de degrés  $k$  est stocké à l'indice  $k$  dans la liste. Le polynôme nul est représenté par la liste vide : `[]`.

1. Écrire une fonction `degre(P)` qui prend en paramètre un polynôme  $P$  et qui renvoie son degré.
2. Écrire une fonction `somme(P, Q)` qui prend en paramètre deux polynômes  $P$  et  $Q$  et qui renvoie leur somme. Attention le dernier élément ne doit pas être nul.
3. Écrire une fonction `produit(P, Q)` qui prend en paramètre deux polynômes  $P$  et  $Q$  et qui renvoie leur produit.
4. Écrire une fonction `evaluation(P, x)` qui prend en paramètre un polynôme  $P$  et un flottant  $x$  qui renvoie  $P(x)$ .
5. Écrire une fonction `puissance(P, n)` qui prend en paramètre un polynôme  $P$  et un entier nature  $n$  qui renvoie  $P^n$ .

---

**Si vous avez fini :**

Dans le thème matrices, ajouter des fonctions pour permuter deux lignes d'une matrice et opérer une transvection. Coder alors l'algorithme du pivot partiel (une recherche sur le web pourra aider).

Dans le thème polynômes, coder l'algorithme de division euclidienne (même conseil).