

## TD 4 : Les matrices

---

### Exercice N° 1 :

Ecrire une fonction « **unitaire(n)** » qui construit et retourne une matrice *carrée unitaire* de dimension **n**. Une matrice unitaire est une matrice, telle que :

$$u_{ij} = \begin{cases} 1 & \text{si } i=j \\ 0 & \text{si } i \neq j \end{cases}$$

```
def unitaire(n):  
    M = [[0]*n for i in range(n)]  
    for i in range(n):  
        M[i][i] = 1  
    return M
```

### Exercice N° 2 :

Ecrire une fonction « **transpose( M )** » qui retourne la transposition  $M^t$  d'une matrice M.

$$\begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix}^t = \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix}$$

```
def transpose(M):  
    l,c = len(M), len(M[0])  
    T = [[0]*l for i in range(c)]  
    for i in range(c):  
        for j in range(l):  
            T[i][j] = M[j][i]  
    return T
```

### Exercice N° 3 :

Ecrire une fonction « **somme(A, B)** » qui retourne la somme de deux matrices A et B de mêmes dimensions.

```
def somme(A, B):
    l, c = len(A), len(A[0])

    S = [[0]*c for i in range(l)]

    for i in range(l):
        for j in range(c):
            S[i][j] += A[i][j] + B[i][j]

    return S
```

#### **Exercice N° 4 :**

En multipliant une matrice A de dimensions M et N avec une matrice B de dimensions N et P on obtient une matrice C de dimensions M et P:  $A(M,N) * B(N,P) = C(M,P)$

La multiplication de deux matrices se fait en multipliant les composantes des deux matrices lignes par colonnes :

$$c_{ij} = \sum_{k=1}^N (a_{ik} * b_{kj})$$

Ecrire une fonction « **produit(A, B)** » qui retourne le produit des deux matrices A et B.

```
def produit(A, B):
    M, N = len(A), len(A[0])
    P = len(B[0])

    C = [[0]*P for i in range(M)]

    for i in range(M):
        for j in range(P):
            for k in range(N):
                C[i][j] += A[i][k] * B[k][j]

    return C
```

#### **Exercice N° 5 :** (Trace d'une matrice carrée)

Écrire une fonction trace(A) d'argument A une matrice et qui retourne la somme des coefficients diagonaux de A (appelée trace de A) si A est une matrice carrée et la chaîne de caractères « trace of a non square matrix is not defined » sinon.

```
def trace(A):
    if len(A) != len(A[0]):
        return "trace of a non square matrix is not defined"

    return sum(A[i][i] for i in range(len(A)))
```

**Exercice N° 6 :**

Écrire une fonction **is\_upper\_triangular(A)** qui retourne True si A est une matrice triangulaire supérieure (donc en particulier carrée), et False sinon.

```
def is_upper_triangular(A):
    if len(A) != len(A[0]):
        return False

    n = len(A)
    for i in range(1, n):
        for j in range(i):
            if A[i][j] != 0:
                return False

    return True
```