

EXERCICE 1 — PRODUIT DE DEUX MATRICES CARRÉES

On rappelle qu'en Python, les matrices peuvent être représentées par des objets de type array. Pour définir de tels objets, il suffit d'importer la bibliothèque numpy, avec l'instruction :

```
import numpy as np
```

Ceci fait, on peut définir la matrice $A = \begin{pmatrix} 1 & 2 & 4 \\ 7 & 8 & 9 \end{pmatrix}$ par l'instruction :

```
A = np.array([[1,2,4],[7,8,9]])
```

On accède à un coefficient de la matrice A avec une "paire de crochets". Explicitement, avec la matrice A définie ci-dessus, on a par exemple :

```
A[0][0] = 1; A[0][2] = 4; A[1][1] = 8; A[1][2] = 9; A[2][1] = error
```

Par ailleurs, on peut définir une matrice nulle à n lignes et p colonnes grâce à l'instruction : `np.zeros([n,p])`. Explicitement, l'instruction `A = np.zeros([2,4])` définit A comme la matrice nulle de $M_{2,4}(\mathbb{R})$:

```
A = array([[0., 0., 0., 0.],
           [0., 0., 0., 0.]])
```

QUESTION Ecrire en Python le code d'une fonction `PROD(A,B,n)` qui reçoit comme paramètres deux matrices carrées A et B à n lignes et n colonnes, et qui retourne le produit AB de ces deux matrices.

```
1 # SOLUTION 1
```

```
2
3 def PROD(A,B,n):
4     C = np.zeros([n,n]) # Initialisation de la variable C à la matrice nulle de M_n(R)
5     for i in range(n):
6         for j in range(n):
7             S = 0
8             for k in range(n):
9                 S = S + A[i][k] * B[k][j]
10            C[i][j] = S
11     return C
```

```
12 #####
```

```
13
14 # SOLUTION 2
```

```
15
16 def PROD(A,B,n):
17     C = np.zeros([n,n]) # Initialisation de la variable C à la matrice nulle de M_n(R)
18     for i in range(n):
19         for j in range(n):
20             for k in range(n):
21                 C[i][j] = C[i][j] + A[i][k] * B[k][j]
22     return C
```

EXERCICE 2 — CRÉATION DE LA MATRICE IDENTITÉ

QUESTION Ecrire en Python le code d'une fonction `IDENT(n)` qui reçoit comme paramètre un entier n , et qui retourne la matrice identité I_n de $M_n(\mathbb{R})$.

```

1  # SOLUTION "NAÏVE"
2
3  def IDENT(n):
4      C = np.zeros([n,n])
5      for i in range(n):
6          C[i][i] = 1
7      return C
8  #####
9
10 # SOLUTION "OH YEAH!"
11
12 def IDENT(n):
13     return np.array([[1*(i==j) for j in range(n)] for i in range(n)])
14
15 # Rque: 1*(i==j) est la représentation en Python du symbole de Kronecker

```

EXERCICE 3 — PUISSANCES D'UNE MATRICE CARRÉE

QUESTION Ecrire en Python le code d'une fonction `PUISS(A,p,n)` qui reçoit comme paramètres une matrice carrée A à n lignes et n colonnes, et un entier p , et qui retourne la matrice A^p .

```

1  def PUISS(A,p,n):
2      C = IDENT(n)
3      for compteur in range(p):
4          C = PROD(C,A,n)
5      return C

```