Interrogation d'Informatique

du 23 au 27/2/2026

Nom :		Prénom :	

Consignes

- En Mathématiques, les énoncés du cours doivent être complètement écrits : un cadre, des hypothèses et une conclusion.
- En Informatique : les scripts doivent être correctement indentés, en mettant en valeur l'indentation à l'aide d'une barre verticale.
- La note finale tiendra compte, directement ou indirectement, de la qualité de la rédaction et de la présentation.
- **Le crayon à papier ne sera pas corrigé.**
- **L'usage de la calculatrice est interdit.**

/20

TABLEAUX

Exercice 1 | [Solution] /3 Soit $A \in \mathfrak{M}_{n,n}(\mathbb{R})$ définie par :

$$\forall (i,j) \in \{1,\dots n\}^2, \quad A_{ij} = i^2 + (j-1)^2.$$

Écrire une fonction `creer_matrice(n)` renvoyant un tableau `numpy` correspondant à A.

Exercice 2 | [Solution] Pour tout $n \geqslant 1$. On considère la matrice $K_n \in \mathfrak{M}_{n+1,n+1}(\mathbb{R})$ telle que :

- $\forall i \in [\![1\,,\,n]\!], \quad (K_n)_{i,i+1} = i,$
- $\forall j \in [\![1\,,\,n]\!], \quad (K_n)_{j+1,j} = -n-1+j,$
- et dont tous les autres coefficients sont nuls.

Le premier point définit donc la sur-diagonale de la matrice, le second la sous-diagonale. On a donc par exemple :

$$K_1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad \text{et} \quad K_2 = \begin{pmatrix} 0 & 1 & 0 \\ -2 & 0 & 2 \\ 0 & -1 & 0 \end{pmatrix}.$$

/3 Écrire une fonction `matriceK(n)` en Python qui prend en entrée $n \in \mathbb{N}^*$ et qui renvoie la matrice K_n.

Exercice 3 | [Solution] Dans cet exercice, on considère que M est une matrice de $\mathcal{M}_{n,p}(\mathbb{R})$ et que i et j sont deux entiers vérifiant $(i,j) \in [\![1\,,\,n]\!] \times [\![1\,,\,p]\!]$. On s'interdira dans cet exercice l'utilisation de toute fonction `sum` présente dans Python ou dans le module `numpy`, et on notera bien que l'exercice **ne suppose pas les matrices carrées**.

1. /2 Écrire une fonction `sommeL(M, i)` qui renvoie la somme des éléments de la ligne d'indice i de M.

2. /2 Écrire une fonction `produit(M)` qui renvoie le produit des éléments de M.

SUITES

Exercice 4 | [Solution] Soit (u_n) la suite définie par :

$$u_0 = 1, \quad \text{et :} \quad \forall n \in \mathbb{N}, \quad u_{n+1} = \frac{\sin(u_n)}{2}.$$

On admet que (u_n) converge vers 0. *Dans tout l'exercice, on pourra importer le module de son choix, afin d'accéder à la fonction sinus. Cette importation devra être réalisée dans la première question, et on la supposera réalisée dans la suite.*

1. /3 Écrire une fonction **non-récursive** `terme_u(n)` qui renvoie la valeur u_n.

2. /2 Écrire une fonction **récursive** `terme_u_rec(n)` qui renvoie la valeur u_n.

3. /3 Écrire une fonction `list_u(n)` qui renvoie la liste $[u_0, \dots, u_n]$.

4. [/2] Écrire une fonction `vitesse_u(e)` qui renvoie le premier indice n pour lequel : $|u_n| < e$.

SOLUTIONS DES EXERCICES

Solution (exercice 1) [Énoncé]

```
def creer_matrice(n):
    A = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
                A[i, j] = (i+1)**2 + j**2
    return A
```

```
>>> creer_matrice(3)
array([[ 1.,  2.,  5.],
       [ 4.,  5.,  8.],
       [ 9., 10., 13.]])
```

Solution (exercice 2) [Énoncé]

```
def matriceK(n):
    K = np.zeros((n+1, n+1))
    for i in range(n):
        K[i, i+1] = i+1
    for j in range(n):
        K[j+1, j] = -n-1+(j+1)
    return K
```

```
>>> matriceK(1)
array([[ 0.,  1.],
       [-1.,  0.]])
>>> matriceK(2)
array([[ 0.,  1.,  0.],
       [-2.,  0.,  2.],
       [ 0., -1.,  0.]])
```

Solution (exercice 3) [Énoncé]

1.
```
def sommeL(M, i):
    S = 0
    p = len(M[0])
    for j in range(p):
        S += M[i-1, j] # i est un indice Maths
    return S
```

```
>>> M = np.array([[1, 2], [3, 4]])
>>> sommeL(M, 1) # somme de la ligne 1 (indice maths)
np.int64(3)
```

2.
```
def produit(M):
    P = 1
    n, p = len(M), len(M[0])
    for i in range(n):
        for j in range(p):
            P *= M[i, j]
    return P
```

```
>>> produit(M)
np.int64(24)
```

Solution (exercice 4) [Énoncé]

1.
```
import math as ma
def terme_u(n):
    if n == 0:
        return 1
    else:
        u = 1
        for _ in range(1, n+1):
            u = ma.sin(u)/2
        return u
```

```
>>> terme_u(3)
0.1013997354308852
```

2.
```
def terme_u_rec(n):
    if n == 0:
        return 1
    else:
        return ma.sin(terme_u_rec(n-1))/2
```

```
>>> terme_u_rec(3)
0.1013997354308852
```

3. Écrire une fonction `list_u(n)` qui renvoie la liste de tous les termes de u_0 à u_n.

```
def list_u(n):
    if n == 0:
        return [1]
    else:
        L = [1]
        for i in range(1, n+1):
```

```
            L.append(ma.sin(L[-1])/2)
        return L
```

```
>>> list_u(10)
[1, 0.42073549240394825, 0.20421595634496814, 0.10139973543088
52, 0.050613030488148844, 0.0252957121003766, 0.01264650725630
2666, 0.00632308507880458, 0.0031615214722923346, 0.0015807581
028061176, 0.0007903787222370743]
```

4. Écrire une fonction `vitesse_u(e)` qui renvoie le premier indice n pour lequel $|u_n| < e$.

```
def vitesse_u(e):
    u = 1
    while abs(u) >= e:
        u = ma.sin(u)/2
    return u
```

```
>>> vitesse_u(0.001)
0.0007903787222370743
```