Exercice 1

Dans tout l'exercice, la lettre n désigne un entier naturel non nul.
On considère la fonction f_n définie par :

$$\forall x \in [0,1],\ f_n(x) = \sum_{k=1}^{n} kx^k = x + 2x^2 + 3x^3 + \ldots + nx^n$$

1. 1.a. Montrer que f_n est strictement croissante sur $[0,1]$.
   1.b. En déduire que l'équation $f_n(x) = 1$, d'inconnue x, possède une seule solution, notée u_n, élément de $[0,1]$.
   1.c. Donner la valeur de u_1.
2. 2.a. Pour tout réel x de $[0,1]$, exprimer $f_{n+1}(x)$ en fonction de $f_n(x)$.
   2.b. En déduire que $f_{n+1}(u_n) \geqslant 1$.
   2.c. Utiliser les variations de f_{n+1} pour conclure que la suite $(u_n)_{n\in\mathbb{N}^*}$ est décroissante.
   2.d. Montrer que la suite $(u_n)_{n\in\mathbb{N}^*}$ est convergente. On note ℓ sa limite.
3. 3.a. Pour tout réel $x \neq 1$, rappeler la formule donnant $\displaystyle\sum_{k=0}^{n} x^k$ en fonction de x et n.
   3.b. En déduire que, pour tout réel x différent de 1, on a l'égalité :
   $$\sum_{k=1}^{n} kx^{k-1} = \frac{nx^{n+1} - (n+1)x^n + 1}{(1-x)^2}$$
   3.c. Donner alors une expression sans symbole $\sum$ de $f_n(x)$ pour $x \in [0,1[$.
4. 4.a. Déterminer u_2 puis en déduire que, si n est supérieur ou égal à 2 , on a : $0 \leqslant u_n \leqslant \dfrac{1}{2}$.
   4.b. En déduire $\displaystyle\lim_{n\to+\infty} u_n^n$ et $\displaystyle\lim_{n\to+\infty} nu_n^n$.
   4.c. En revenant à la définition de u_n, montrer, pour $n \geqslant 2$, l'égalité :
   $$u_n^2 - 3u_n + 1 = nu_n^{n+2} - (n+1)u_n^{n+1}$$
   4.d. Donner finalement la valeur de ℓ.

Exercice 2

On note E l'ensemble des matrices de la forme $M(a,b) = \begin{pmatrix} a & b & a \\ b & 2a-b & b \\ a & b & a \end{pmatrix}$, où a et b sont des réels.

1. 1.a. Montrer que E est un sous-espace vectoriel de $\mathcal{M}_3(\mathbb{R})$.
   1.b. Donner une base de E et en déduire sa dimension.
2. Justifier sans calcul que les matrices de E sont diagonalisables mais pas inversibles.

Dans toute la suite, sauf la dernière question, on étudie un exemple.

On note I la matrice identité de $\mathcal{M}_3(\mathbb{R})$ et on considère la matrice $A = \begin{pmatrix} 1 & 3 & 1 \\ 3 & -1 & 3 \\ 1 & 3 & 1 \end{pmatrix}$.

3. Vérifier que A appartient à E.
4. Écrire une fonction **Python** d'en-tête `matA()` retournant la matrice A.
5. 5.a. Quelle valeur propre de A la question 2 permet-elle d'obtenir ?
   5.b. Montrer que les matrices $A - 5I$ et $A + 4I$ ne sont pas inversibles. En déduire deux autres valeurs propres de A.
   5.c. Déterminer une base de chaque sous-espace propre de A puis construire une base (U,V,W) de $\mathcal{M}_{3,1}(\mathbb{R})$ formée de vecteurs propres de A (on prendra pour chacun d'entre eux la première composante égale à 1).
6. On considère les instructions Python suivantes :

```python
r1=al.matrix_rank(matA()-5*np.eye(3,3))
r2=al.matrix_rank(matA()+4*np.eye(3,3))
print('r1=',r1)
print('r2=',r2)
```

Utiliser la question précédente pour donner les valeurs de r_1 et r_2 renvoyées par ce script.

7. 7.a. Vérifier que les vecteurs U, V et W sont vecteurs propres de toutes les matrices de E.
   7.b. Soit n un entier naturel non nul. En utilisant la matrice P dont les colonnes sont les vecteurs U, V et W, indiquer comment obtenir la puissance n-ième de n'importe quelle matrice de E (seule la démarche est exigée, les calculs et leurs résultats numériques ne sont pas demandés).
   7.c. En déduire, sans la commande **`al.matrix_power`**, et toujours pour $n \in \mathbb{N}^*$, une fonction **`Python`** d'en-tête **`puissanceM(a, b, n)`** renvoyant $M(a, b)^n$.

Exercice 3

On suppose que les variables aléatoires présentées dans cet exercice sont toutes définies sur un même espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$.
Dans tout l'exercice, la lettre n désigne un entier naturel non nul.

1. Soit f_n la fonction définie par $f_n(x) = \begin{cases} \left(1 - \dfrac{x}{n}\right)^{n-1} & \text{si } 0 \leqslant x \leqslant n \\ 0 & \text{sinon} \end{cases}$.

   Vérifier que f_n est une densité.

Dans la suite, on considère une suite $(X_n)_{n\in\mathbb{N}^*}$ de variables aléatoires telle que, pour tout entier naturel n non nul, X_n admet f_n comme densité.

2. 2.a. Justifier que $\mathbb{E}\left(1 - \dfrac{X_n}{n}\right)$ et $\mathbb{E}\left(\left(1 - \dfrac{X_n}{n}\right)^2\right)$ existent et donner leur expression en fonction de n.
   2.b. En déduire que X_n possède une espérance et une variance et donner leur expression en fonction de n.
3. Déterminer la fonction de répartition F_n de X_n.
4. 4.a. Donner, pour tout réel x strictement négatif, la limite de $F_n(x)$ quand n tend vers $+\infty$.
   4.b. Soit x un réel positif. Montrer que, pour tout entier $n \geqslant \lfloor x \rfloor + 1$, on a :
   $$F_n(x) = 1 - \left(1 - \frac{x}{n}\right)^n$$
   4.c. Pour tout réel x positif, calculer $\lim\limits_{n\to\infty} n \ln\left(1 - \dfrac{x}{n}\right)$.
   4.d. Déduire des questions précédentes que la suite $(X_n)_{n\in\mathbb{N}^*}$ converge en loi vers une variable X dont on donnera la loi.
5. Soient $U_1, \ldots, U_n$ des variables aléatoires mutuellement indépendantes, et suivant toutes la loi uniforme sur $[0, 1]$. On considère la variable aléatoire M_n définie par $M_n = \min(U_1, \ldots, U_n)$, ce qui signifie que, pour tout $\omega \in \Omega$, $M_n(\omega)$ est le plus petit des réels $U_1(\omega), \ldots, U_n(\omega)$.
   Enfin, on pose $Z_n = nM_n$.

   5.a. En notant G la fonction de répartition commune à $U_1, \ldots, U_n$, rappeler l'expression de $G(x)$ selon que $x < 0$, $0 \leqslant x \leqslant 1$ ou $x > 1$.
   5.b. Déterminer, pour tout réel x, la probabilité $\mathbb{P}\big([Z_n > x]\big)$ à l'aide de la fonction G et en déduire explicitement la fonction de répartition F_{Z_n} de Z_n.
   5.c. Conclure que Z_n suit la même loi que X_n.
   5.d. Utiliser la question 5.c pour écrire une fonction **`Python`** renvoyant une réalisation de X_n.

Exercice 4

Dans ce problème, n désigne un entier naturel non nul.
On dispose de $n+1$ urnes, numérotées de 1 à $n+1$, et contenant chacune n boules.
Pour tout k de $[\![1; n+1]\!]$, l'urne numéro k contient $k-1$ boules noires, les autres boules étant blanches (ainsi, l'urne numérotée 1 ne contient que des boules blanches et l'urne numérotée $n+1$ ne contient que des boules noires). L'épreuve consiste à choisir une urne au hasard et à y effectuer indéfiniment des tirages au hasard d'une boule, avec remise de la boule tirée dans l'urne dont elle provient après chaque tirage.
Pour tout k de $[\![1; n+1]\!]$, on note U_k l'événement : "On a choisi l'urne numérotée k".
On appelle X_n la variable aléatoire qui prend la valeur 0 si l'on n'obtient aucune boule blanche au cours de l'épreuve et qui prend la valeur j ($j \in \mathbb{N}^*$) si la première boule blanche apparaît au j-ième tirage.

Pour finir, on rappelle les commandes Python suivantes qui permettent de simuler certaines variables discrètes usuelles : **`rd.randint(a,b+1)`** simule une variable aléatoire suivant la loi uniforme sur $[\![a, b]\!]$.
`rd.binomial(n,p)` simule une variable aléatoire suivant la loi binomiale de paramètres n et p.
`rd.geometric(p)` simule une variable aléatoire suivant la loi géométrique de paramètre p.

1. Simulation de X_n : pour tout j de $[\![2; n+1]\!]$, on code les $j-1$ boules noires de l'urne numérotée j par les entiers de $[\![1, j-1]\!]$. Compléter alors la fonction **`Python`** suivante pour qu'elle renvoie la valeur prise par X_n lors de l'épreuve aléatoire décrite ci-dessus :

```python
import numpy.random as rd

def varX(n):
    k=... #choix de l'urne
    if k==n+1:
        X=...
    elif k==1:
        X=...
    else:
        X=1
        while rd.randint(1,n+1)<=...:
            X=...
    return (X)
```

2. Pour tout k de $[\![1;n+1]\!]$, déterminer $\mathbb{P}(U_k)$.
3. 3.a. Pour tout k de $[\![1;n]\!]$, donner la loi de X_n, conditionnellement à l'événement U_k.
   3.b. En conservant, sans les écrire de nouveau, les 6 premières lignes de la fonction **Python** précédente, compléter les 3 lignes suivantes afin d'obtenir une nouvelle simulation de X_n :

```python
else:
    X=...
return (X)
```

4. 4.a. Déterminer $\mathbb{P}_{U_{n+1}}\big([X_n=1]\big)$.
   4.b. Pour tout k de $[\![1;n]\!]$, donner $\mathbb{P}_{U_k}\big([X_n=1]\big)$.
   4.c. Montrer alors que $\mathbb{P}\big([X_n=1]\big)=\dfrac{1}{2}$.
5. Soit j un entier supérieur ou égal à 2.
   5.a. Déterminer $\mathbb{P}_{U_{n+1}}\big([X_n=j]\big)$.
   5.b. Pour tout k de $[\![1;n]\!]$, donner $\mathbb{P}_{U_k}\big([X_n=j]\big)$.
   5.c. En déduire l'égalité :
$$\mathbb{P}\big([X_n=j]\big)=\frac{1}{n+1}\sum_{k=0}^{n-1}\left[\left(\frac{k}{n}\right)^{j-1}-\left(\frac{k}{n}\right)^{j}\right]$$
6. 6.a. Justifier que, pour tout k de $[\![0;n-1]\!]$, on a : $\displaystyle\sum_{j=2}^{+\infty}\left[\left(\frac{k}{n}\right)^{j-1}-\left(\frac{k}{n}\right)^{j}\right]=\frac{k}{n}$.
   6.b. Calculer $\mathbb{P}\big([X_n\geqslant 2]\big)$ en fonction de n.
7. 7.a. Déduire des questions précédentes l'expression de $\mathbb{P}\big([X_n=0]\big)$ en fonction de n.
   7.b. Aurait-on pu anticiper ce dernier résultat sans aucun calcul ?
8. 8.a. Montrer que X_n possède une espérance $\mathbb{E}(X_n)$ donnée par :
$$\mathbb{E}(X_n)=\frac{n}{n+1}\sum_{p=1}^{n}\frac{1}{p}$$
   8.b. Informatique : calcul et affichage de $\mathbb{E}(X_n)$.
   Compléter le script suivant afin qu'il permette de calculer et d'afficher $\mathbb{E}(X_n)$:

```python
n=int(input('entrez la valeur de n :'))
v=np.arange(1,n+1)
E=...
print(E)
```

9. 9.a. Montrer que : $\forall p\in\mathbb{N}^*,\ \dfrac{1}{p+1}\leqslant\displaystyle\int_p^{p+1}\frac{1}{t}dt\leqslant\frac{1}{p}$.
   9.b. En déduire, pour tout n de $\mathbb{N}^*\setminus\{1\}$, l'encadrement : $\displaystyle\sum_{p=2}^{n}\frac{1}{p}\leqslant\ln(n)\leqslant\sum_{p=1}^{n-1}\frac{1}{p}$.
   9.c. Établir enfin l'encadrement :
$$\forall n\in\mathbb{N}^*\setminus\{1\},\ \ln(n)+\frac{1}{n}\leqslant\sum_{p=1}^{n}\frac{1}{p}\leqslant\ln(n)+1$$
   9.d. Utiliser l'encadrement précédent pour donner l'équivalent le plus simple possible de $\mathbb{E}(X_n)$ lorsque n est au voisinage de $+\infty$.

★★★★★★★ FIN ★★★★★★★