Concours Blanc n°1
Maths 1
27/11/2023
Durée : 4h

La présentation, la lisibilité, l'orthographe, la qualité de la rédaction, la clarté et la précision des raisonnements entreront pour une part importante dans l'appréciation des copies.
Les candidats sont invités à ***encadrer*** *dans la mesure du possible les résultats de leurs calculs.*
Ils ne doivent faire usage d'aucun document. ***L'utilisation de toute calculatrice et de tout matériel électronique est interdite.*** *Seule l'utilisation d'une règle graduée est autorisée.*
Si au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il la signalera sur sa copie et poursuivra sa composition en expliquant les raisons des initiatives qu'il sera amené à prendre.

Exercice 1

On considère l'espace vectoriel $\mathcal{M}_2(\mathbb{R})$ des matrices carrées d'ordre 2 à coefficients réels.
On définit :

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \qquad C = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \qquad T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

et

$$\mathcal{E} = \left\{ \begin{pmatrix} a & b \\ 0 & c \end{pmatrix} \;\middle|\; (a,b,c) \in \mathbb{R}^3 \right\}$$

1. Montrer que $\mathcal{E}$ est un espace vectoriel et que $\{A, B, C\}$ est une base de $\mathcal{E}$.
   Quelle est la dimension de $\mathcal{E}$?

2. Établir que $\mathcal{E}$ est stable par multiplication, c'est-à-dire : $\forall\, (M, N) \in \mathcal{E}^2,\ MN \in \mathcal{E}$.

3. Montrer que, pour toute matrice M de $\mathcal{E}$, si M est inversible, alors $M^{-1} \in \mathcal{E}$.

Pour toute matrice $M \in \mathcal{E}$, on note $f(M) = TMT$.

4. Montrer que f est un endomorphisme de $\mathcal{E}$.

5. Calculer $f(A), f(B), f(C)$ en fonction de A,B,C et en déduire $\mathrm{Mat}\big(f, \{A, B, C\}\big)$. On note cette matrice F.

6. Résoudre l'équation $FX = X$, où $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathcal{M}_{3,1}(\mathbb{R})$.

7. En déduire les solutions de l'équation $f(M) = M$, d'inconnue $M \in \mathcal{E}$.

On note $I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ et $H = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$.

8. Calculer H^2 ; puis H^n pour $n \geqslant 2$.

9. Calculer, pour tout $a \in \mathbb{R}$ et tout $n \in \mathbb{N}$, $(I + aH)^n$.

10. En déduire la valeur de F^n pour tout $n \in \mathbb{N}$.

11. Trouver une matrice $G \in \mathcal{M}_3(\mathbb{R})$ telle que $G^3 = F$.
    Existe-t-il un endomorphisme g de $\mathcal{E}$ tel que $g \circ g \circ g = f$?

Exercice 2 : Bases de données

On s'intéresse ici à une table de données cinématographiques.
On veut manipuler :

- Une table `Acteurs`, avec les colonnes suivantes :
  - `id_acteur`, un numéro identifiant l'acteur
  - `Nom`
  - `Pays_acteur`
  - `Age`
  - `Sexe` (vaudra `M` ou `F`)
  - `Taille` (exprimée en cm)
- Une table `Films`, avec les colonnes suivantes :
  - `id_film`, un numéro identifiant le film
  - `Titre`
  - `Acteur_principal`, qui contiendra le numéro identifiant l'acteur dans la table `Acteurs`
  - `Année`
  - `Durée` (exprimée en minutes)
  - `Pays_Film`

1. Quelles clés primaires et clés étrangères doit-on déclarer sur ces tables ? (on précisera vers quoi doivent pointer les clés étrangères).
2. On suppose la table `Acteurs` créée. Donner une syntaxe SQL permettant de créer la table `Films`, en tenant compte de votre réponse à la question précédente (vous déclarerez pour chaque colonne des formats de données qui vous semblent pertinents).
3. Donner une requête SQL permettant de lister les titres des films français réalisés avant 1980.
4. Donner une requête SQL permettant de calculer la durée moyenne des films finlandais.
5. Donner une requête SQL permettant d'afficher les titres et durées des films français et des films japonais, triés par ordre chronologique, du plus ancien au plus récent.

On utilise maintenant les deux tables.

6. Écrire la jointure de ces deux tables, avec la condition qui semble pertinente.

Pour simplifier les écritures suivantes, on pourra noter « J » pour le code qui répond à la question précédente.

7. Donner une requête SQL permettant d'afficher les titres et années des films finlandais dont l'acteur principal est français.
8. Donner une requête SQL permettant d'afficher le nombre de films qui ont un acteur principal de sexe féminin ?

On veut maintenant enrichir les informations, en répertoriant plusieurs acteurs qui jouent dans un même film.
On adopte pour cela le modèle suivant :

- La table `Acteurs` est inchangée.
- Dans la table `Films`, on retire la colonne `Acteur_principal`.
- On crée une nouvelle table `JoueDans`, contenant 2 colonnes `Acteur` et `Film`, dont chaque enregistrement signale qu'un acteur joue dans un film. Dans cette table, acteurs et films sont représentés par leurs identifiants des tables précédentes.

Par exemple, si cette table contient les lignes :

Acteur	Film
⋮	⋮
1	2
3	2
1	4
⋮	⋮

cela signifie que l'acteur tel que `Acteurs.id_acteur = 1` joue dans le film tel que `Films.id_film = 2` ; que l'acteur 3 joue également dans le film 2 ; etc...

9. La table `JoueDans` contient-elle des clés primaires ?
10. La table `JoueDans` contient-elle des clés étrangères ? si oui, vers quoi pointent-elles ?
11. Donner les noms de tous les acteurs jouant dans des films français. On fera en sorte d'évier qu'un même acteur apparaisse plusieurs fois dans les résultats.
Pour cela on aura à effectuer une double jointure pour relier les 3 tables ; la syntaxe consiste à enchaîner à la suite deux commandes de jointure :

`Table1` [jointure avec `Table2` avec la condition] [jointure avec `Table3` avec la condition]

Exercice 3

Soit n un entier naturel non nul.
On effectue une série illimitée de tirages d'une boule avec remise dans une urne contenant n boules numérotées de 1 à n. Pour tout entier naturel k non nul, on note X_k la variable aléatoire égale au numéro de la boule obtenue au k-ième tirage.
Pour tout entier naturel k non nul, on note S_k la somme des numéros des boules obtenues lors des k premiers tirages :

$$S_k = \sum_{i=1}^{k} X_i.$$

On considère enfin la variable aléatoire T_n égale au nombre de tirages nécessaires pour que, pour la première fois, la somme des numéros des boules obtenues soit supérieure ou égale à n (T_n est donc le plus petit entier $k \in \mathbb{N}^*$ tel que $S_k \geqslant n$).

Exemple : avec $n = 10$, si les numéros obtenus aux cinq premiers tirages sont dans cet ordre $2, 4, 1, 5, 9$, alors on obtient : $S_1 = 2$, $S_2 = 6$, $S_3 = 7$, $S_4 = 12$, $S_5 = 21$ et $T_{10} = 4$.

Partie A

1. Pour $k \in \mathbb{N}^*$, déterminer la loi de X_k ainsi que son espérance.
2. (a) Déterminer $T_n(\Omega)$.
   (b) Calculer $P(T_n = 1)$.
   (c) Montrer que
   $$P(T_n = n) = \left(\frac{1}{n}\right)^{n-1}$$
3. Dans cette question, $n = 2$. Déterminer la loi de T_2.
4. Dans cette question, $n = 3$. Donner la loi de T_3. Vérifier que $E(T_3) = \dfrac{16}{9}$.
5. Compléter la fonction Python ci-dessous, qui prend en argument le nombre n de boules contenues dans l'urne, afin qu'elle simule la variable aléatoire T_n:

```python
import numpy as np
import numpy.random as rd
def T(n):
    S = ..........
    t = ..........
    while .......... :
        tirage = ....
        S = S + tirage
        t = ..........
    return t
```

Partie B

6. Déterminer $S_k(\Omega)$ pour tout $k \in \mathbb{N}^*$.

7. Soit $k \in [\![1, n-1]\!]$.

   (a) Exprimer S_{k+1} en fonction de S_k et de X_{k+1}.

   (b) En utilisant un système complet d'événements lié à la variable aléatoire S_k, démontrer alors que :

$$\forall i \in [\![k+1, n]\!],\ P(S_{k+1} = i) = \frac{1}{n} \sum_{j=k}^{i-1} P(S_k = j).$$

8. (a) Pour $k \in \mathbb{N}^*$ et $j \in \mathbb{N}^*$ tels que $k \leqslant j$, rappeler (sans démonstration) la formule du triangle de Pascal liant les nombres : $\binom{j-1}{k-1}$, $\binom{j-1}{k}$ et $\binom{j}{k}$. On pourra utiliser la convention $\binom{k}{k+1} = 0$.

   (b) En déduire que pour tout $k \in \mathbb{N}^*$ et pour tout entier naturel i supérieur ou égal à $k+1$:

$$\sum_{j=k}^{i-1} \binom{j-1}{k-1} = \binom{i-1}{k}.$$

   (c) Pour tout entier $k \in [\![1, n]\!]$, on note $\mathscr{P}_k$ la proposition :

$$\text{« } \forall i \in [\![k, n]\!],\ P(S_k = i) = \frac{1}{n^k}\binom{i-1}{k-1} \text{ »}$$

   Démontrer par récurrence que pour tout entier $k \in [\![1, n]\!]$, $\mathscr{P}_k$ est vraie.

9. (a) Soit $k \in [\![1, n-1]\!]$. Justifier soigneusement que les événements $[T_n > k]$ et $[S_k \leqslant n-1]$ sont égaux.

   (b) En déduire que: $\forall k \in [\![0, n]\!],\ P(T_n > k) = \dfrac{1}{n^k}\dbinom{n-1}{k}$.

10. Démontrer que $E(T_n) = \displaystyle\sum_{k=0}^{n-1} P(T_n > k)$, puis que $E(T_n) = \left(1 + \dfrac{1}{n}\right)^{n-1}$.

11. Calculer $\displaystyle\lim_{n\to+\infty} E(T_n)$.

Partie C

12. Montrer que : $\forall\, k \in \mathbb{N}^*$, $\displaystyle\lim_{n\to+\infty} P(T_n > k) = \frac{1}{k!}$, et en déduire $\displaystyle\lim_{n\to+\infty} P(T_n = k)$.

13. On code la fonction `mystere` suivante :

```python
def mystere(n):
    L=np.zeros(8)
    for k in range(10000):
        tirage=T(n)
        if tirage <= 8:
            L[tirage-1]=L[tirage-1]+1
    return L/10000
```

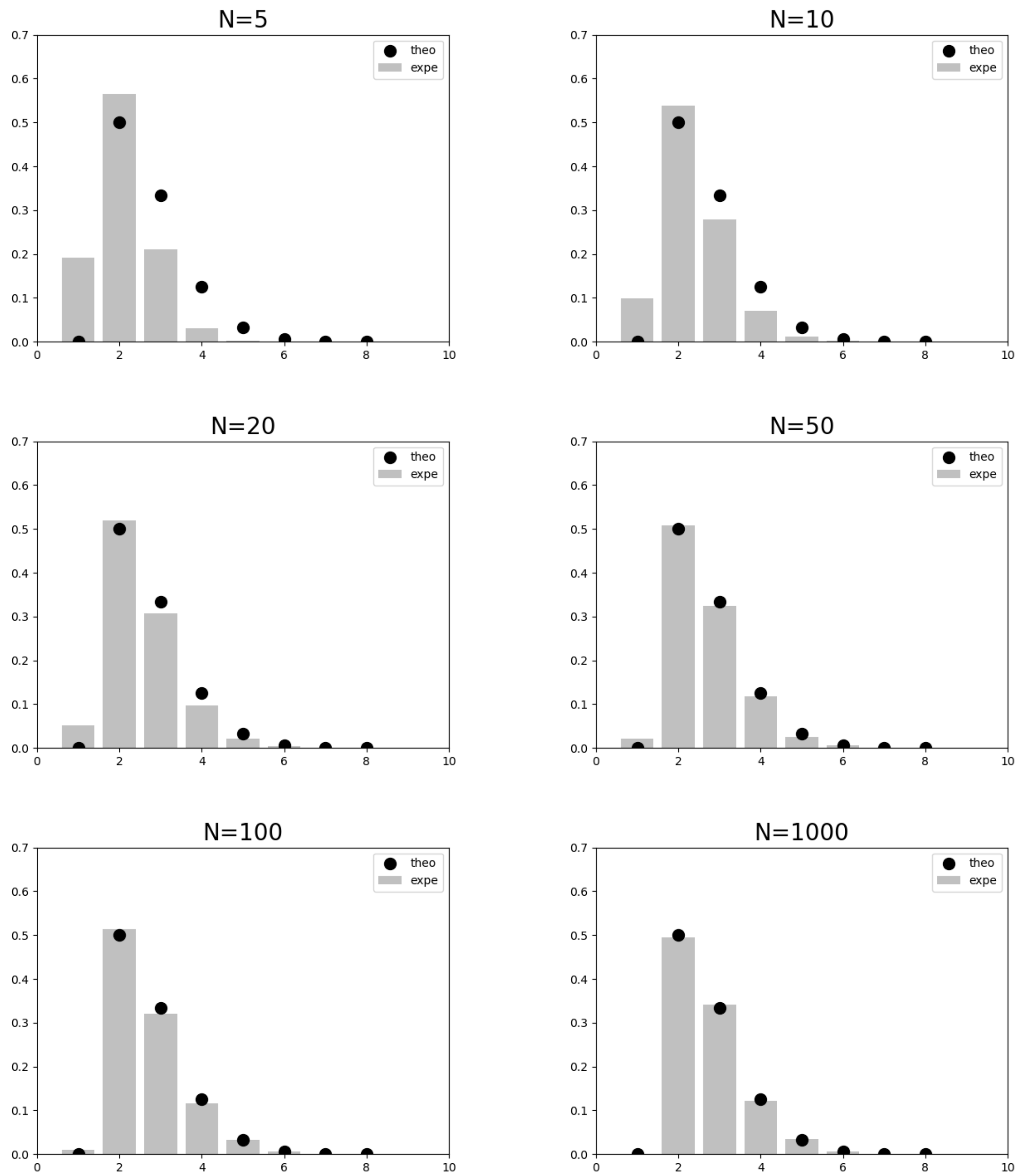
On définit aussi les listes :

```python
expe=mystere(N)
theo=[(k-1)/np.math.factorial(k) for k in range(1,9)]
```

(`np.math.factorial` est la fonction factorielle).
Enfin on trace un graphique par la commande suivante :

```python
plt.bar(range(1,9),expe,label ='expe')
plt.scatter(range(1,9),theo,label ='theo')
plt.title('N='+str(N))
plt.show()
```

Pour diverses valeurs de N, on obtient les figures suivantes :



Expliquer ce que fait la fonction `mystere`, et les figures obtenues.

Exercice 4

Pour tout entier naturel n, on pose $u_n = \prod_{k=0}^{n}\left(1+\frac{1}{2^k}\right) = (1+1)\left(1+\frac{1}{2}\right)\left(1+\frac{1}{4}\right)\cdots\left(1+\frac{1}{2^n}\right)$.

1. Donner, sous forme d'entiers ou de fractions simplifiées, les valeurs de u_0, u_1 et u_2.
2. Donner une suite d'instructions Python permettant de calculer et d'afficher u_{50}.
3. (a) Montrer que, pour tout entier naturel n, on a : $u_n \geqslant 2$.
   (b) Exprimer u_{n+1} en fonction de u_n puis en déduire les variations de la suite $(u_n)_{n\in\mathbb{N}}$.
4. (a) Établir que, pour tout réel x strictement supérieur à -1, on a : $\ln(1+x) \leqslant x$.
   (b) En déduire un majorant de la suite $\big(\ln(u_n)\big)_{n\in\mathbb{N}}$.
5. En utilisant les questions précédentes, montrer que la suite $(u_n)_{n\in\mathbb{N}}$ converge vers un réel ℓ, élément de $[2, e^2]$.
6. On se propose dans cette question de déterminer la nature de la série de terme général $(\ell - u_n)$.
   (a) Justifier que la suite $(\ln(u_n))_{n\in\mathbb{N}}$ converge et que l'on a : $\ln(\ell) = \sum_{k=0}^{+\infty} \ln\left(1+\frac{1}{2^k}\right)$.
   (b) Montrer que, pour tout n de $\mathbb{N}$, on a $\ln\left(\frac{\ell}{u_n}\right) = \sum_{k=n+1}^{+\infty} \ln\left(1+\frac{1}{2^k}\right)$.
   (c) Vérifier, en utilisant le résultat de la question 4a, que $\forall n \in \mathbb{N},\ 0 \leqslant \ln\left(\frac{\ell}{u_n}\right) \leqslant \frac{1}{2^n}$.
   (d) Déduire de la question précédente que $\forall n \in \mathbb{N},\ 0 \leqslant \ell - u_n \leqslant \ell\left(1 - e^{-\frac{1}{2^n}}\right)$.
   (e) Justifier que, pour tout réel x, on a $1 - e^{-x} \leqslant x$. En déduire que $\forall n \in \mathbb{N},\ 0 \leqslant \ell - u_n \leqslant \frac{\ell}{2^n}$.
   Conclure quant à la nature de la série de terme général $(\ell - u_n)$.