DS 8 - Concours blanc
Durée 3H30

- Les calculatrices sont interdites durant les cours, TD et *a fortiori* durant les DS de mathématiques.
- Si vous pensez avoir découvert une erreur, indiquez-le clairement sur la copie et justifiez les initiatives que vous êtes amenés à prendre.
- Une grande attention sera apportée à la clarté de la rédaction et à la présentations des solutions. (Inscrivez clairement en titre le numéro de l'exercice, vous pouvez aussi encadrer les réponses finales.)
- Vérifiez vos résultats.
- Le résultat d'une question peut être admis et utilisé pour traiter les questions suivantes en le signalant explicitement sur la copie.
- Une liste des fonctions Python disponibles se trouvent à la fin du sujet.

Exercice 1. Les questions de cet exercice sont indépendantes.

1. Justifier que $E=\{(x,y,z)\in\mathbb{R}^3 \mid xyz=0\}$ n'est pas un sous-espace vectoriel de $\mathbb{R}^3$.
2. Soit $F=\{(x,y,z,t)\in\mathbb{R}^4 \mid x+y+z+t=0, 2x+y+z-t=0\}$ Montrer que F est sous-espace vectoriel de $\mathbb{R}^4$ et en donner une base.
3. Montrer que $((1,2,3),(1,0,1),(2,2,1))$ est une base de $\mathbb{R}^3$.
4. Donner les DL à l'ordre 2 en 0 de $\exp(x), \ln(\cos(x)), \sqrt{1+2x}$ et $\sin(x^2)$ puis en déduire la limite suivante
$$\lim_{x\to 0}\frac{e^x+\ln(\cos(x))-\sqrt{1+2x}}{\sin(x^2)}$$

Exercice 2 (Agro 2016)**.** On considère la suite $(S_n)_{n\in\mathbb{N}^*}$ définie par

$$\forall n\in\mathbb{N}^*,\quad S_n=\sum_{k=1}^{n}\frac{\ln k}{k}.$$

1. Ecrire une fonction Python `suiteS` qui prend en argument un entier `n` et retourne la valeur de S_n.
2. Étude de la nature de la suite $(S_n)_{n\in\mathbb{N}^*}$.
   (a) Dresser le tableau de variations de la fonction $f: x\mapsto \frac{\ln(x)}{x}$.
   (b) Ecrire un script Python qui permet de tracer et d'afficher le graphe de f entre 1 et 10 pour l'axe des abscisses et -10 et 10 pour l'axe des ordonnées. Sur l'axe des abscisses on prendra 100 points par intervalle de taille 1.
   (c) En déduire que, pour tout entier k supérieur ou égal à 4 , on a :
$$\int_k^{k+1}\frac{\ln(x)}{x}\,\mathrm{d}x\leq\frac{\ln(k)}{k}\leq\int_{k-1}^{k}\frac{\ln(x)}{x}\,\mathrm{d}x$$
   (d) En déduire l'existence de trois constantes réelles positives A, B et C telles que, pour tout entier naturel n supérieur ou égal à 4 , on ait :
$$\frac{\ln^2(n+1)}{2}-A\leq S_n-B\leq\frac{\ln^2(n)}{2}-C.$$
   (e) En déduire la limite de la suite $(S_n)_{n\in\mathbb{N}^*}$.
3. Recherche d'un équivalent de S_n.
   (a) Montrer que $\ln^2(n+1)\underset{n\to+\infty}{\sim}\ln^2(n)$.
   (b) En déduire que $S_n\underset{n\to+\infty}{\sim}\frac{\ln^2(n)}{2}$.
4. Étude asymptotique de la suite u définie par :
$$\forall n\in\mathbb{N}^*,\quad u_n=S_n-\frac{\ln^2(n)}{2}$$
Soit $g(n)=\frac{\ln^2(n)}{2}$. On note $\tau_{n,n+1}(g)=\frac{g(n+1)-g(n)}{n+1-n}$. On admet le résultat [1] suivant : $\forall n\in\mathbb{N}^*$
$$\inf_{x\in[n,n+1]}g'(x)\leq\tau_{n,n+1}(g)\leq\sup_{x\in[n,n+1]}g'(x)$$
   (a) Montrer à l'aide du résultat admis que, pour tout entier n supérieur ou égal à 3, $u_{n+1}-u_n\leq 0$.

1. Appelé inégalité des accroissements finis

(b) En déduire que la suite u converge. On note ℓ sa limite.

(c) Conclure que $S_n \underset{+\infty}{=} \frac{\ln^2(n)}{2} + \ell + o(1)$

Exercice 3 (D'après Agro 2019)**.** On considère l'expérience suivante : on effectue une suite de lancers d'une piéce truquée dont la probabilité d'obtenir face vaut $p \in [0,1]$. On suppose les lancers indépendants. Pour tout entier $n \in \mathbb{N}^*$, on notera F_n l'événément :

$$F_n = \{\text{Face est obtenu au n-ème lancer}\}$$

1. Pour tout $n \in \mathbb{N}^*$, on note E_n l'événement

$$E_n = \{\text{Le premier face est obtenu au n-ème lancer}\}$$

   (a) Pour tout $n \in \mathbb{N}^*$, exprimer E_n en fonction des $(F_k)_{k\in[\![1,n]\!]}$.

   (b) En déduire $P(E_n)$ en fonction de n et p.

2. Pour tout $n \in \mathbb{N}^*$, on note A_n l'événement

$$A_n = \{\text{Le premier face est obtenu avant (pas strictement) le n-ème lancer}\} \quad \text{et} \quad B_n = \overline{A_n}$$

   (a) Soit $n \in \mathbb{N}^*$. Exprimer B_n en fonction des $(F_k)_{k\in[\![1,n]\!]}$.

   (b) En déduire $P(B_n)$ en fonction de n et p.

   (c) Soit $(n,m) \in (\mathbb{N}^*)^2$, montrer que

$$P_{B_n}(B_{n+m}) = P(B_m).$$

3. On suppose dans cette question que la piéce est équilibrée, c'est-à-dire que $p = \frac{1}{2}$.

   On appelle "double face" l'obtention du coté face deux fois consécutivement. Par exemple la suite $(face, pile, face, face, pile)$ est une suite de 5 tirages avec un double face, ce dernier est obtenu au 4eme lancer.
   Pour tout $n \in \mathbb{N}^*$, on note Q_n l'événement

$$Q_n = \{\text{Suite de } n \text{ lancers sans double face}\}$$

   On note $q_n = P(Q_n)$

   (a) Calculer q_1 et q_2.

   (b) Justifier que

$$Q_{n+2} = \{(pile, \omega_{n+1}) \,|\, \omega_{n+1} \in Q_{n+1}\} \cup \{(face, pile, \omega_n) \,|\, \omega_n \in Q_n\}$$

   (c) En déduire que pour tout $n \in \mathbb{N}^*$

$$q_{n+2} = \frac{1}{2}q_{n+1} + \frac{1}{4}q_n$$

   (d) Déterminer les racines du polynôme $X^2 - \frac{1}{2}X - \frac{1}{4}$. On les notera r_1 et r_2 avec $r_1 < r_2$.

   (e) Justifier que la matrice $\begin{pmatrix} r_1 & r_2 \\ r_1^2 & r_2^2 \end{pmatrix}$ est inversible.

(f) En déduire qu'il existe $(A, B) \in \mathbb{R}^2$ tel que

$$\begin{cases} Ar_1 + Br_2 & = & q_1 \\ Ar_1^2 + Br_2^2 & = & q_2 \end{cases}$$

(On ne demande pas de déterminer explicitement A et B)

(g) Prouver que pour tout $n \in \mathbb{N}^*$, $q_n = Ar_1^n + Br_2^n$.

4. INFO - On revient au cas général où la pièce est truquée et la probabilité d'obtenir face vaut $p \in [0, 1]$

   (a) Ecrire une fonction Python `tirage` qui prend en argument un flottant `p` et retourne `'F'` avec probabilité p et `'P'` avec probabilité $1 - p$.

   (b) Ecrire une fonction Python `suite_tirage` qui prend en argument un flottant `p` et un entier `n` et qui retourne une liste de `n` éléments dont chaque entrée correspond à un tirage de pièce comme défini dans la question précédente.

   (c) Ecrire une fonction Python `nombre_face` qui prend en argument un flottant `p` et un entier `n` et qui retourne le nombre de fois où face a été obtenu. (on n'utilisera pas la fonction `count` de Python)

   (d) Ecrire une fonction Python `premier_face` qui prend en argument un flottant `p` correspondant à la probabilité d'obtenir 'Face' et retourne le numero du tirage du premier face.

   (e) Ecrire une fonction Python `premier_double_face` qui prend en argument un flottant `p` correspondant à la probabilité d'obtenir 'Face' et retourne le numero du tirage du premier double face.

Liste (non-exhaustive) des fonctions utilisables.

Bibliothéque `random`

```
import random as rd
rd.random() #renvoie un flottant aleatoire entre 0 et 1
rd.randint(a,b) #renvoie un entier aleatoire entre a et  b inclus
rd.choice(L) #renvoie un element aleatoirement dans une liste L
```

Bilbiothéque `math`

```
import math as m
m.log(x) # renvoie ln(x) si x>0
m.exp(x) # renvoie exp(x)
m.floor(x) # renvoie la partie entirere de x
m.sqrt(x) # renvoie la racine carre de x.
```

Bibliothéque `matplotlib`

```
import matplotlib.pyplot as plt
plt.plot(X,Y, '-') # Genere  la courbe des points definis par les listes
                 X et Y (abscisses et ordonnees)
plt.bar(X,Y) # Genere l'histogramme des points definis par les listes
                X et Y (abscisses et ordonnees)
plt.axis('equal') #rend le repere orthonorme
plt.xlim(xmin, xmax) # fixe les bornes de l'axe des abscisses entre
                xmin et xmax
plt.ylim(ymin, ymax) # fixe les bornes de l'axe des abscisses entre
                ymin et ymax
plt.show() # affiche le graphique
```