CONCOURS BLANC ÉPREUVE I

ECG2 MATHS APPLIQUÉES

Exercice 1 **EML** 2025 Exercice 1.

Partie A : Étude de la suite $(u_n)_{n\in\mathbf{N}}$.

On s'intéresse à la suite récurrente $(u_n)_{n\in\mathbf{N}}$ définie par $u_0 = 1$ et

$$\forall n \in \mathbf{N}, \quad u_{n+1} = u_n \mathrm{e}^{1/u_n}.$$

1. **a.** Montrer que $u_n > 0$ pour tout $n \in \mathbf{N}$.

b. Donner le sens de variation de la suite $(u_n)_{n\in\mathbf{N}}$.

c. Démontrer, en raisonnant par l'absurde, que $(u_n)_{n\in\mathbf{N}}$ admet $+\infty$ comme limite.

2. Recopier et compléter le programme Python ci-dessous de sorte qu'il affiche le premier entier $n \in \mathbf{N}$ tel que $u_n \geqslant 10^6$.

```python
import numpy as np
u = 1
n = 0
while ... :
    u = ...
    n = ...
print(...)
```

Partie B : Étude de la fonction f.

On considère la fonction f définie sur $]0, +\infty[$ par :

$$\forall x > 0, \quad f(x) = x\mathrm{e}^{1/x}$$

On note $\mathscr{C}_f$ la courbe de f dans le plan muni d'un repère orthonormé.

3. Calculer les limites de f en $+\infty$ et en 0 .

4. Dresser le tableau de variation de f sur $]0, +\infty[$.

5. Soit $x > 0$.

a. Justifier la convergence de la série $\sum_{k\geqslant 0} \frac{x^{-k}}{k!}$ et calculer sa somme.

b. En déduire que :

$$f(x) = x + 1 + \frac{1}{x}\sum_{k=2}^{+\infty} \frac{x^{2-k}}{k!}$$

6. Soit $x \geqslant 1$.

a. Établir séparément les inégalités suivantes :

$$\frac{1}{2} \leqslant \sum_{k=2}^{+\infty} \frac{x^{2-k}}{k!} \leqslant \mathrm{e}$$

---

http://louismerlin.fr.

b. En déduire que :

$$(*) \qquad \frac{1}{2x} \leqslant f(x) - (x+1) \leqslant \frac{\mathrm{e}}{x}.$$

7. Montrer que $f(x) = x + 1 + o(1)$ au voisinage de $+\infty$.

8. Représenter sur un même dessin la courbe $\mathscr{C}_f$ et la droite d'équation $y = x + 1$.

Partie C : Comportement asymptotique de la suite $(u_n)_{n\in\mathbf{N}}$**.**

9. a. Montrer que, pour tout entier $k \in \mathbf{N}$, $\quad \ln(u_{k+1}) - \ln(u_k) = \frac{1}{u_k}$.

b. En déduire que, pour tout entier $n \in \mathbf{N}^*$, $\quad \ln(u_n) = \sum\limits_{k=0}^{n-1} \frac{1}{u_k}$.

10. a. À l'aide de l'encadrement (*) montrer que, pour tout $k \in \mathbf{N}$,

$$1 + \frac{1}{2u_k} \leqslant u_{k+1} - u_k \leqslant 1 + \frac{\mathrm{e}}{u_k}$$

b. Soit $n \in \mathbf{N}^*$, établir :

$$n + \frac{1}{2}\sum_{k=0}^{n-1} \frac{1}{u_k} \leqslant u_n - 1 \leqslant n + \mathrm{e}\sum_{k=0}^{n-1} \frac{1}{u_k}$$

puis

$$1 + \frac{1}{2}\ln(u_n) \leqslant u_n - n \leqslant 1 + \mathrm{e}\ln(u_n)$$

11. a. Justifier que : $\lim_{n\to+\infty} \frac{\ln(u_n)}{u_n} = 0$.

b. En déduire un équivalent simple de u_n lorsque n tend vers $+\infty$.

12. Déterminer un équivalent simple de $\sum\limits_{k=0}^{n-1} \frac{1}{u_k}$ lorsque n tend vers $+\infty$.

EXERCICE 2 **EML** 2025 Exercice 2.

Les deux parties de cet exercice sont indépendantes l'une de l'autre.

PARTIE A : RÉDUCTION SIMULTANÉE ET SPECTRE

Soit $\mathscr{M}_3(\mathbf{R})$ l'espace vectoriel des matrices carrées d'ordre trois à coefficients réels. On pose :

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad J = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad \text{et} \quad K = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

et on considère $\mathscr{E} = \mathrm{Vect}(I, J, K)$ le sous-espace vectoriel de $\mathscr{M}_3(\mathbf{R})$ engendré par les matrices I, J et K.

1. Montrer que (I, J, K) est une base de $\mathscr{E}$, en déduire la dimension de $\mathscr{E}$.

2. Justifier sans calcul que les matrices J et K sont diagonalisables.

3. a. Exprimer la matrice J^3 comme un multiple de J.

b. En déduire que les valeurs propres de J appartiennent à l'ensemble $\{-\sqrt{2}, 0, \sqrt{2}\}$.

On pose $U_1 = \begin{pmatrix} \sqrt{2} \\ 1 \\ 1 \end{pmatrix}$ et $U_2 = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$.

4. **a.** Vérifier que U_1 et U_2 sont des vecteurs propres de J.

b. Déterminer un vecteur propre U_3 de J associé à la valeur propre $-\sqrt{2}$.

5. **a.** Justifier que (U_1, U_2, U_3) est une base de $\mathscr{M}_{3,1}(\mathbf{R})$.

b. Donner une matrice inversible P de $\mathscr{M}_3(\mathbf{R})$ telle que :

$$P^{-1}JP = \begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\sqrt{2} \end{pmatrix}$$

6. **a.** Montrer que (U_1, U_2, U_3) est aussi une base de vecteurs propres de K.

b. Déterminer la matrice $P^{-1}KP$.

7. Soit M une matrice de $\mathscr{E}$ de coordonnées $(a, b, c) \in \mathbf{R}^3$ dans la base (I, J, K).

a. Exprimer la matrice $P^{-1}MP$ sous la forme d'un tableau de nombres dépendant de a, b et c.

b. En déduire les valeurs propres de M.

8. On considère l'application linéaire $s : \mathscr{E} \to \mathbf{R}^3$ définie par :

$$s(M) = s(aI + bJ + cK) = (a + b\sqrt{2} + c, a - c, a - b\sqrt{2} + c)$$

pour toute matrice $M = aI + bJ + cK$ avec $(a, b, c) \in \mathbf{R}^3$.

a. Donner la matrice S de s relativement à la base (I, J, K) de $\mathscr{E}$ et à la base canonique de $\mathbf{R}^3$.

b. Montrer que l'application linéaire s est bijective.

Partie B : Un algorithme de coloration des graphes

Soit $n \geqslant 1$ un entier, on considère un graphe non orienté G donné par sa matrice d'adjacence $A \in \mathscr{M}_n(\mathbf{R})$. On note $\mathscr{S} = \{s_0, \ldots, s_{n-1}\}$ l'ensemble des sommets de G, dans les programmes informatiques on confondra un sommet s_i avec son numéro i. On dit que deux sommets sont voisins s'ils sont distincts et reliés par une arête.

Une coloration de G est une application $c : \mathscr{S} \to \mathbf{N}$ telle que $c(s_i) \neq c(s_j)$ si les sommets s_i et s_j sont voisins. Dans cette définition, $\mathbf{N}$ représente l'ensemble des "couleurs" disponibles, la coloration c attribue à chaque sommet une "couleur" de sorte que deux sommets voisins soient de "couleurs" différentes.

Le graphe G admet la coloration triviale donnée par $c(s_i) = i$ pour tout $i \in [\![0, n-1]\!]$, il peut cependant admettre une coloration nécessitant moins de n "couleurs". Ainsi, le graphe à cinq sommets ci-dessous admet la coloration à trois "couleurs" définie par : $c(s_0) = 0, c(s_1) = 1, c(s_2) = 0, c(s_3) = 1, c(s_4) = 2$.

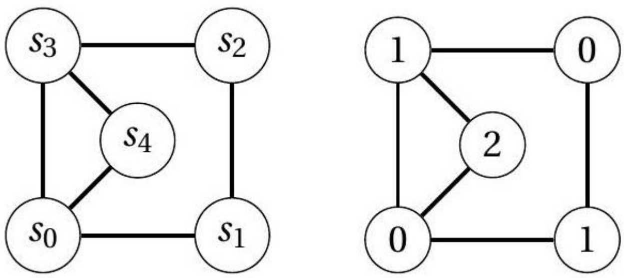

FIGURE 1. Un graphe d'ordre cinq colorié avec trois "couleurs"

Les questions suivantes ont pour but de réaliser un programme Python qui renvoie une coloration d'un graphe G quelconque, en essayant de minimiser le nombre de couleurs utilisées. On commence par rédiger deux fonctions auxiliaires, `voisins` et `min_ext`, qui serviront pour la fonction finale `coloration`. On suppose que la matrice d'adjacence A de G est définie à l'aide de la commande `np.array`.

9. Recopier et compléter le programme Python ci-dessous de manière à ce qu'il définisse une fonction voisins, prenant en arguments la matrice d'adjacence A et un entier $i \in [\![0, n-1]\!]$, et renvoyant la liste des sommets voisins de s_i.

```
def voisins(A,i):
    n = len(A[i])
    V = []
    for j in range(n):
        if j!= i and ... :
            V.append(...)
    return(V)
```

10. Rédiger en Python une fonction min_ext qui prend en argument une liste d'entiers naturels L, et qui renvoie le plus petit entier naturel n'appartenant pas à L (par exemple, si $L = [1, 0, 3,]$ alors la commande min_ext(L) renvoie 2). On pourra transcrire en langage Python l'algorithme suivant :

On affecte à une variable m la valeur 0.
Tant que m appartient à la liste L :
    On augmente de 1 la valeur de m.
On renvoie m.

11. À l'aide des fonctions introduites précédemment on rédige maintenant une fonction coloration prenant en argument la matrice d'adjacence $A \in \mathscr{M}_n(\mathbf{R})$ d'un graphe G, et renvoyant une coloration de G sous la forme d'une liste d'entiers $C = [C_0, \ldots, C_{n-1}]$, où C_i désigne la "couleur" du sommet s_i pour tout $i \in [\![0, n-1]\!]$.

On construit cette fonction selon l'algorithme "glouton" ci-dessous :

On affecte à la variable n le nombre de sommets de G.
On affecte à la variable C la liste [0,1, ..., n-1].
Pour i allant de 1 à n-1 :
    On affecte à la variable C_voisins la liste des «couleurs» des sommets voisins de s_i.
    On affecte à C_i le plus petit entier naturel qui n'est pas élément de la liste C_voisins.
On renvoie la liste C.

Recopier et compléter la fonction coloration ci-dessous.

```
def coloration(A):
    n = len(A[0])
    C = ...
    for i in range(1,n):
        C_voisins = [ ... for j in ...]
        C[i] = min_ext(...)
    return(C)
```

12. On note A la matrice d'adjacence du graphe G représenté en figure 2 ci-dessous.

a. Donner la liste obtenue en exécutant la commande coloration(A).

b. Le graphe G admet-il une coloration à trois couleurs ? Si oui, exhiber une telle coloration.

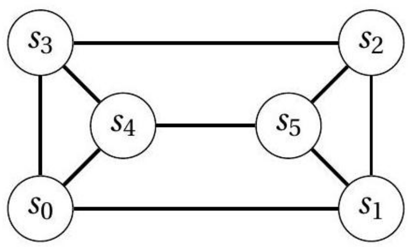

FIGURE 2. Le graphe G

Exercice 3

Dans tout l'exercice, p désigne un réel de $]0,1[$ et on pose $q = 1 - p$.

Toutes les variables aléatoires considérées dans cet exercice sont supposées définies sur le même espace de probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$.

On considère une variable aléatoire X à valeurs dans $\mathbb{N}$, dont la loi est donné par,

$$\forall k \in \mathbb{N}, \quad \mathbb{P}([X = k]) = q^k p = (1-p)^k p.$$

Partie A.

1. Montrer que la variable aléatoire $Y = X + 1$ suit une loi géométrique dont on précisera le paramètre.

2. En déduire que X admet une espérance et une variance et préciser $\mathbb{E}(X)$ et $\mathbb{V}(X)$.

3. Compléter la fonction Python suivante afin que, prenant en entrée le réel p, elle renvoie une simulation de la variable aléatoire X.

```
def simul_X(p) :
  Y = ....
  while .... :
    Y = Y+1
  X = Y-1
  return X
```

Partie B.

Un casino a conçu une nouvelle machine dont le fonctionnement est le suivant :

- le joueur introduit un nombre k de jetons de son choix ($k \in \mathbb{N}$), puis il appuie sur un bouton pour activer la machine ;
- si k est égal à 0, alors la machine ne reverse aucun jeton au joueur ;
- si k est un entier supérieur ou égal à 1, alors la machine définit k variables aléatoires $X_1, \cdots, X_k$, toutes indépendantes et de même loi que la variable aléatoire X étudiée dans la partie A, et reverse au joueur $(X_1 + \cdots + X_k)$ jetons ;
- les fonctionnements de la machine à chaque activation sont indépendants les uns des autres et ne dépendent que du nombre de jetons introduits.

Le casino s'interroge sur la valeur à donner à p pour que la machine soit attractive pour le joueur, tout en étant rentable.

Le casino imagine alors le cas d'un joueur invétéré qui, avant chaque activation, place l'intégralité de ses jetons dans la machine et continue de jouer encore et encore.

On note, pour tout n de $\mathbb{N}$, Z_n la variable aléatoire égale au nombre de jetons dont dispose le joueur après n activations de la machine.

On suppose que le joueur commence avec un seul jetons ; ainsi $Z_0 = 1$.

On remarque en particulier que Z_1 suit la même loi que X.

4. Compléter la fonction Python suivante afin que, prenant en entrée un entier n de $\mathbb{N}$ et le réel p, elle simule l'expérience aléatoire et renvoie la valeur de Z_n.

Cette fonction devra utiliser la fonction simul_X.

```
def simul_Z(n,p) :
  Z = 1
  for i in range(1,n+1) :
    s = 0
    for j in range(1,Z+1) :
      ....
    Z = ....
  return Z
```

On définit, pour tout n de $\mathbb{N}$, u_n la probabilité que le joueur n'ait plus de jetons après n activations de la machine. Ainsi, pour tout $n \in \mathbb{N}$, $u_n = \mathbb{P}([Z_n = 0])$.

On note également R l'événement : "le joueur finit par ne plus avoir de jetons".

5. **a.** Préciser les valeurs de u_0 et u_1.

b. Comparer, pour tout $n \in \mathbb{N}$, les événements $[Z_n = 0]$ et $[Z_{n+1} = 0]$.
En déduire que la suite $(u_n)_{n\in\mathbb{N}}$ est monotone et convergente.

Dans toute la suite de l'exercice, on note $\ell = \lim\limits_{n\to+\infty} u_n$.

6. Justifier que $\mathbb{P}(R) = \ell$.

7. **a.** Montrer que pour tout k de $\mathbb{N}$, on a

$$\mathbb{P}_{[Z_1=k]}([Z_2 = 0]) = (u_1)^k.$$

On **admet** que, pour tout n de $\mathbb{N}$, on a

$$\mathbb{P}_{[Z_1=k]}([Z_{n+1} = 0]) = (u_n)^k.$$

b. En déduire que, pour tout $n \in \mathbb{N}$,

$$u_{n+1} = \sum_{k=0}^{+\infty} \mathbb{P}([Z_1 = k])(u_n)^k = \frac{p}{1 - qu_n}.$$

8. Montrer que ℓ vérifie $(\ell - 1)(q\ell - p)$.

9. On suppose $p \geqslant \frac{1}{2}$. Montrer que $\mathbb{P}(R) = 1$.

10. On suppose $p < \frac{1}{2}$. Montrer que pour tout $n \in \mathbb{N}$, $u_n \in \left[0, \frac{p}{q}\right]$. En déduire que $\mathbb{P}(R) < 1$.

11. Expliquer pourquoi le casino préférera choisir p dans l'intervalle $\left[\frac{1}{2}, 1\right]$.

Partie C.

On suppose à présent que $p \geqslant \frac{1}{2}$.

Le casino cherche la valeur à donner à p pour que le joueur joue le plus longtemps possible dans le casino et ainsi, dépense plus d'argent dans les consommations au bar.

On note T la variable aléatoire égale au nombre d'activations de la machine effectuées par le joueur lorsque, pour la première fois, celui-ci n'a plus de jeton.

On pose, pour tout n de $\mathbb{N}$, $v_n = 1 - u_n$.

12. Justifier que pour tout $n \in \mathbb{N}$, $u_n = \mathbb{P}([T \leqslant n])$ puis que, pour tout $n \in \mathbb{N}$, $\mathbb{P}([T = n]) = v_{n-1} - v_n$.

13. Montrer que pour tout N de $\mathbb{N}^*$,

$$\sum_{n=1}^{N} n\mathbb{P}([T = n]) = \sum_{n=0}^{N-1} v_n - Nv_N.$$

14. On suppose dans cette question que $p = \frac{1}{2}$.

a. Montrer que pour tout $n \in \mathbb{N}$, $u_n = \frac{n}{n+1}$.

b. En déduire que la variable aléatoire T n'admet pas d'espérance.

15. On suppose maintenant que $p > \frac{1}{2}$.

On pose, pour tout n de $\mathbb{N}$, $w_n = \frac{1-u_n}{\frac{p}{q}-u_n}$.

a. Montrer que pour tout $n \in \mathbb{N}$, $w_{n+1} = \frac{q}{p} w_n$.

b. En déduire que pour tout $n \in \mathbb{N}$,

$$u_n = \frac{1 - \left(\frac{q}{p}\right)^n}{1 - \left(\frac{q}{p}\right)^{n+1}}$$

puis que, pour tout $n \in \mathbb{N}$,

$$0 \leqslant v_n \leqslant \left(\frac{q}{p}\right)^n.$$

c. Montrer que la variable aléatoire T admet une espérance et que l'on a $\mathbb{E}(T) \leqslant \frac{1}{1-\frac{q}{p}}$.

16. Quelle(s) valeur(s) de p recommanderiez-vous au casino ?