Dans les questions faisant intervenir des instructions en langage **Python**, on prendra soin d'importer les bibliothèques nécessaires lors de leur première utilisation.
Pour traiter les questions d'informatique, les candidats sont invités à se référer aux annexes fournies en fin de sujet. Ils ne sont pas limités à l'utilisation des seules fonctions mentionnées dans ces annexes.

Exercice 1

Pour toute variable aléatoire X à valeurs dans $\mathbb{N}^*$, et pour tout entier k, on pose :

$$R_X(k) = \mathbb{P}\big([X > k]\big)$$

La partie II est indépendante des autres parties ; les résultats de la partie I pourront intervenir dans la partie III.

Partie I

1. Soit p un réel de $]0;1[$. Dans cette question uniquement, on suppose que X suit la loi géométrique de paramètre p.
   - 1.a. Calculer $R_X(k)$ pour tout entier naturel k.
   - 1.b. Vérifier que : $\forall k \in \mathbb{N}^*,\ \dfrac{R_X(k)}{R_X(k-1)} = 1 - p$.
2. Soient X et Y deux variables aléatoires discrètes à valeurs dans $\mathbb{N}^*$.
   - 2.a. Pour tout entier naturel non nul k, exprimer $\mathbb{P}\big([X = k]\big)$ à l'aide de la fonction R_X.
   - 2.b. En déduire que X et Y suivent la même loi si, et seulement si, pour tout entier naturel k, $R_X(k) = R_Y(k)$.

Partie II

3. 3.a. Déterminer deux réels a et b tels que : $\forall n \in \mathbb{N}^*,\ \dfrac{n}{(n+1)!} = \dfrac{a}{n!} - \dfrac{b}{(n+1)!}$.
   - 3.b. En déduire que la série $\displaystyle\sum_{n \geqslant 1} \frac{n}{(n+1)!}$ est convergente et déterminer la valeur de la somme $\displaystyle\sum_{n=1}^{+\infty} \frac{n}{(n+1)!}$.
4. Dans cette question, on considère une variable aléatoire X à valeurs dans $\mathbb{N}^*$ telle que :

$$\forall n \in \mathbb{N}^*,\ \mathbb{P}\big([X = n]\big) = \frac{n}{(n+1)!}$$

   - 4.a. Montrer que la variable aléatoire $X + 1$ admet une espérance et calculer $\mathbb{E}(X+1)$.
     En déduire que X admet une espérance et calculer $\mathbb{E}(X)$.
   - 4.b. Montrer que la variable aléatoire $(X-1)(X+1)$ admet une espérance et calculer $\mathbb{E}\big((X-1)(X+1)\big)$.
     En déduire que X admet une variance et calculer $\mathbb{V}(X)$.

Partie III

Soit $(\alpha_k)_{k \in \mathbb{N}^*}$ une suite de réels strictement compris entre 0 et 1.
On étudie la durée de vie en années d'un appareil. Tout au long de l'année initiale $k = 0$, on suppose que l'appareil fonctionne. Puis, à l'issue de chaque année numéro k (k étant un entier naturel non nul), l'appareil possède une certaine probabilité de tomber en panne.
Plus précisément, on suppose que, pour tout entier naturel k non nul, si la machine fonctionne encore à l'issue de la $(k-1)$-ième année, alors elle cesse de fonctionner à la fin de l'année k avec la probabilité α_k, et elle continue à fonctionner après la fin de l'année k avec probabilité $1 - \alpha_k$.
On note X la variable aléatoire égale à la durée de vie en années de l'appareil.

5. Justifier que, pour tout entier naturel k non nul : $R_X(k) = (1 - \alpha_k) R_X(k-1)$.
6. En déduire, pour tout entier naturel k non nul : $R_X(k) = \displaystyle\prod_{i=1}^{k} (1 - \alpha_i)$.
7. En déduire, pour tout entier naturel k non nul, une expression de $\mathbb{P}\big([X = n]\big)$ en fonction des termes de la suite $(\alpha_i)_{i \in \mathbb{N}^*}$. *On pourra utiliser le résultat de la question 2.a.*
8. **Étude de deux exemples.**
   - 8.a. Dans cette question uniquement, on suppose que la suite $(\alpha_k)_{k \in \mathbb{N}^*}$ est constante, c'est-à-dire : $\forall k \in \mathbb{N}^*,\ \alpha_k = p$.
     Reconnaitre la loi de X.
   - 8.b. Dans cette question uniquement, on suppose que, pour tout entier naturel k non nul, $\alpha_k = \dfrac{k}{k+1}$.
     Déterminer la loi de X.

Partie IV

Un fabricant d'ordinateurs souhaite publier des données statistiques sur la durée de vie de ses appareils fabriqués à partir de l'an 2000. Dans une base de données, on dispose d'une table **ordinateur** contenant des informations sur tous les ordinateurs produits par le fabricant. Cette table possède les attributs (ou colonnes) suivants.

- **id** (de type INTEGER) : le numéro d'identification de l'ordinateur.
- **annee_fabrication** (de type INTEGER) : l'année de fabrication de l'ordinateur.
- **adresse_ip** (de type INTEGER) : l'adresse IP associée à l'ordinateur.
- **annee_panne** (de type INTEGER) : l'année où l'ordinateur a cessé de fonctionner, valant -1 si l'ordinateur est encore en état de marche.

Dans les questions qui suivent, en plus des commandes SQL au programme, on pourra utiliser les fonctions présentées dans l'**Annexe B** en fin de sujet.

9. 9.a. Écrire une requête SQL permettant de déterminer le nombre total d'ordinateurs produits par le fabricant.

   9.b. Écrire une requête SQL permettant de déterminer le nombre d'ordinateurs ayant cessé de fonctionner exactement un an après leur production.

   9.c. Dans cette question uniquement, on suppose que la durée de vie en années d'un ordinateur est une variable aléatoire de loi géométrique, de paramètre p inconnu.
   Expliquer de quelle manière le résultat des requêtes écrites dans les questions 9.a et 9.b peut être utilisé pour estimer le paramètre p.

10. Un attribut **duree_vie**, de type INTEGER, a été ajouté à la table **ordinateur**. Aux champs de l'attribut **duree_vie** a été affectée la valeur -1.
    Écrire une requête SQL permettant de modifier la table **ordinateur** en affectant, pour chaque ordinateur, sa durée de vie à l'attribut **duree_vie**.
    Dans le cas des ordinateurs qui sont encore en état de marche, on ne modifiera pas la valeur -1 déjà affectée.

11. Dans cette question, on cherche à déterminer s'il est raisonnable de représenter la durée de vie d'un ordinateur par une variable aléatoire de loi géométrique d'un certain paramètre p que l'on cherchera à approcher.

    11.a. Expliquer comment le résultat de la requête suivante permet d'obtenir une valeur approchée de p.

```
SELECT AVG(duree_vie) FROM ordinateur WHERE duree_vie<>-1
```

    11.b. La base de données compte au total 10000 ordinateurs. On exécute les requêtes suivantes :

```
SELECT   COUNT(*)/10000   FROM   ordinateur   WHERE   duree_vie = 1 ;
SELECT   COUNT(*)/10000   FROM   ordinateur   WHERE   duree_vie = 2 ;
   ⋮                        ⋮                   ⋮
SELECT   COUNT(*)/10000   FROM   ordinateur   WHERE   duree_vie = 24 ;
```

    En utilisant les résultats de la question 8, expliquer de quelle manière les données de la table **ordinateur** peuvent être exploitées pour déterminer s'il est raisonnable de représenter la durée de vie d'un ordinateur par une variable aléatoire de loi géométrique.

Exercice 2

Soit a un réel. On considère la fonction I_a définie par :

$$I_a(x) = \int_x^{+\infty} e^{2a(x-t)-t^2}\,dt.$$

On considère également l'intégrale J_a définie par :

$$J_a = \int_0^{+\infty} e^{-2at-t^2}\,dt.$$

Partie I

1. 1.a. Montrer la relation suivante :

$$e^{-2at-t^2} \underset{t\to+\infty}{=} o\left(\frac{1}{t^2}\right).$$

   1.b. En déduire que l'intégrale J_a est convergente.

2. En déduire que la fonction I_a est définie sur $\mathbb{R}$.

3. 3.a. Justifier que

$$\lim_{x\to+\infty} \int_x^{+\infty} e^{-2at-t^2}\,dt = 0.$$

   3.b. Dans cette question uniquement, on suppose que a est positif.
   Montrer que, pour tout réel x :

$$I_a(x) \leqslant \int_x^{+\infty} e^{-t^2}\,dt$$

   3.c. Déduire des deux questions précédentes que, quelle que soit la valeur du réel a : $\lim\limits_{x\to+\infty} I_a(x) = 0$.

Partie II

On considère l'équation différentielle suivante, d'inconnue $y : \mathbb{R} \to \mathbb{R}$ dérivable :

$$y' = 2ay - e^{-x^2}. \tag{1}$$

Dans cette partie, on s'intéresse aux solutions de l'équation (1) qui vérifient $\lim\limits_{x\to+\infty} y(x) = 0$.

On considère l'équation homogène associée à (1) :

$$y' = 2ay. \tag{2}$$

4. Déterminer l'ensemble des solutions de l'équation homogène (2).
5. On considère la fonction F_a définie sur $\mathbb{R}$ par

$$\forall x \in \mathbb{R},\ F_a(x) = \int_0^x e^{-2at-t^2}\,dt.$$

   5.a. Montrer que F_a est dérivable sur $\mathbb{R}$ et, pour tout réel x, déterminer $F_a'(x)$.

   5.b. Montrer que, pour tout réel x,

$$I_a(x) = e^{2ax}\left(J_a - F_a(x)\right).$$

   5.c. En déduire que la fonction I_a est dérivable sur $\mathbb{R}$ et qu'elle est solution de l'équation différentielle (1).

6. Déterminer l'ensemble des solutions de (1).
7. Déterminer l'ensemble des solutions y de (1) telles que $\lim\limits_{x\to+\infty} y(x) = 0$ dans les trois cas suivants :

   7.a. $a < 0$,

   7.b. si $a = 0$,

   7.c. $a > 0$.

On pourra utiliser le résultat de la question 3.c.

Partie III

On considère une variable aléatoire X de loi normale d'espérance $-a$ et de variance $\dfrac{1}{2}$.

8. 8.a. Rappeler l'expression d'une densité de X.

   8.b. Tracer l'allure de sa courbe représentative dans le cas $a = 2$.

9. Soit x un réel.

   9.a. Exprimer $\mathbb{P}\left([X \geqslant x]\right)$ sous forme d'intégrale.

   9.b. En déduire

$$I_a(x) = \sqrt{\pi}e^{2ax+a^2}\,\mathbb{P}\left([X \geqslant x]\right)$$

10. 10.a. Soit Z une variable aléatoire de loi normale centrée réduite.
    Déterminer, en fonction de a, deux réels α et β tels que $\alpha Z + \beta$ suit la même loi que X.

    10.b. Recopier et compléter la fonction **Python** suivante, prenant en arguments d'entrée les réels a et x, pour qu'elle renvoie une estimation de la probabilité $\mathbb{P}\left([X \geqslant x]\right)$.

```python
import numpy as np
import numpy.random as rd
def estim_proba(a,x):
    num=0
    for i in range(10000):
        Z=rd.normal()
        X=...+Z/...
        if ...:
            num=num+1
    return ...
```

11. Écrire une fonction **Python**, nommée **approx_I**, prenant en arguments d'entrée les réels a et x et renvoyant une valeur approchée de $I_a(x)$.

Exercice 3

Partie I

Soit n un entier naturel supérieur ou égal à 2. On considère la matrice carrée d'ordre n dont tous les coefficients diagonaux sont égaux à 0, et dont tous les autres coefficients sont égaux à 1 :

$$M_n = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}$$

On note I_n la matrice identité d'ordre n.

1. **Étude du cas $n = 3$.**

Dans cette question, on considère la matrice $M = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$.

1.a. Justifier que la matrice M est diagonalisable.

1.b. Calculer $(M + I_3)^2$, puis en déduire un polynôme annulateur de M.

1.c. Déterminer les valeurs propres et une base de chaque sous-espace propre de M.

Dans les questions qui suivent, on considère la matrice $P = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{pmatrix}$.

1.d. Montrer que P est inversible et que :

$$P^{-1} = \frac{1}{3}\begin{pmatrix} 1 & -2 & 1 \\ 1 & 1 & -2 \\ 1 & 1 & 1 \end{pmatrix}$$

Dans les questions qui suivent, on pose $D = P^{-1}MP$.

1.e. Déterminer les coefficients de la matrice D.

1.f. Montrer par récurrence que, pour tout entier naturel k, $M^k = PD^kP^{-1}$.

1.g. Soit k un entier naturel. On admet qu'il existe deux réels a_k et b_k tels que $M^k = a_kM + b_kI_3$.
En utilisant les résultats des questions précédentes, déterminer a_k et b_k.

2. **Cas général : n est un entier naturel quelconque supérieur ou égal à 2.**

On considère la matrice J_n carrée d'ordre n dont tous les coefficients sont égaux à 1 :

$$J_n = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & 1 \end{pmatrix}$$

2.a. Montrer que, pour tout entier naturel k non nul, $(J_n)^k = n^{k-1}J_n$.

2.b. Exprimer M_n en fonction de I_n et J_n.

2.c. En déduire, pour tout entier naturel k non nul

$$(M_n)^k = c_kJ_n + (-1)^kI_n$$

où :

$$c_k = \sum_{i=1}^{k} \binom{k}{i} n^{i-1}(-1)^{k-i}$$

2.d. Montrer que, pour tout entier naturel k non nul,

$$c_k = \frac{(n-1)^k + (-1)^{k+1}}{n}$$

où c_k est le réel défini à la question précédente.

2.e. En déduire, pour tout entier naturel k non nul, une expression des coefficients diagonaux et des coefficients non diagonaux de $(M_n)^k$, en fonction de n et de k.

Partie II

Soit n un entier naturel supérieur ou égal à 2. On considère un graphe non orienté K_n à n sommets numérotés de 1 à n, dans lequel chaque sommet est relié à chaque autre sommet par une arête et n'est pas relié à lui-même par une arête.

3. Représenter graphiquement les graphes K_2, K_3, K_4 et K_5.

4. 4.a. Déterminer la matrice d'adjacence du graphe K_n.

4.b. Dans le graphe K_4, combien existe-t-il de chaînes (ou chemins) de longueur 4 menant du sommet numéro 1 à lui-même ?
On pourra utiliser le résultat de la question 2.e.

5. Déterminer le degré de chaque sommet du graphe K_n.

6. Montrer que le nombre total d'arêtes du graphe K_n est égal à $\dfrac{n(n-1)}{2}$.

Partie III

Soient n un entier naturel supérieur ou égal à 2 et K_n le graphe défini dans la partie II. On parcourt les sommets du graphe K_n de la façon suivante :

- Initialement, à l'étape $k = 0$, on se trouve sur le sommet numéro 1.
- À chaque étape, on change de sommet en suivant au hasard, avec équiprobabilité, l'une des arêtes issues du sommet actuel.

Pour tout entier naturel k, on note X_k la variable aléatoire égale au numéro du sommet sur lequel on se trouve à la k-ième étape (c'est-à-dire à l'issue du k-ième déplacement). En particulier, X_0 est une variable aléatoire constante égale à 1.
Pour tout entier naturel k, on note V_k la matrice ligne de $\mathcal{M}_{1,n}(\mathbb{R})$ définie par :

$$V_k = \begin{pmatrix} \mathbb{P}([X_k = 1]) & \mathbb{P}([X_k = 2]) & \cdots & \mathbb{P}([X_k = n]) \end{pmatrix}$$

7. Déterminer V_0 et V_1.

8. Déterminer la matrice de transition de la chaîne de Markov $(X_k)_{k\in\mathbb{N}}$.

9. 9.a. Rappeler la définition d'un état stable de la chaîne de Markov $(X_k)_{k\in\mathbb{N}}$.

   9.b. Soit V la matrice ligne de $\mathcal{M}_{1,n}(\mathbb{R})$ dont tous les coefficients sont égaux à $\dfrac{1}{n}$:

$$V = \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix}$$

   Montrer que V est un état stable de la chaîne de Markov $(X_k)_{k\in\mathbb{N}}$.

10. 10.a. Pour tout entier naturel k, rappeler sans démonstration une expression de V_{k+1} en fonction de V_k, M_n et n, où M_n est la matrice définie en introduction de la partie I.

    10.b. En déduire, pour tout entier naturel k :

$$V_k = \frac{1}{(n-1)^k} V_0 (M_n)^k$$

    10.c. En utilisant le résultat de la question 2.e, en déduire que la suite $(X_k)_{k\in\mathbb{N}}$ converge en loi vers une variable aléatoire dont on reconnaitra la loi.

11. Comparer et commenter les résultats des questions 9.b et 10.c.

Annexe A - Fonctions Python utiles

Opérations arithmétiques.

- L'opération `//`, appliquée entre deux entiers naturels non nuls, renvoie le quotient de la division euclidienne du premier entier par le second. Par exemple, `13//4` renvoie `3` car $13 = 4 \times 3 + 1$.
- L'opération `%`, appliquée entre deux entiers naturels non nuls, renvoie le reste de la division euclidienne du premier entier par le second. Par exemple, `13%4` renvoie `1` car $13 = 4 \times 3 + 1$.

La bibliothèque numpy.

- Exemple d'importation : `import numpy as np`.
- Les opérations +, `-`, `*`, `/`, `**`, lorsqu'elles sont possibles, peuvent être réalisées entre deux tableaux numpy de dimensions compatibles et agissent alors **coefficient par coefficient**.
- Les fonctions `np.sqrt` (racine carrée), `np.abs` (valeur absolue), `np.log` (logarithme népérien) et `np.exp` (exponentielle) s'appliquent à une quantité numérique ou à un tableau numpy de nombres. Dans ce dernier cas, les fonctions sont appliquées à chaque élément du tableau donné en argument d'entrée.
- Une valeur approchée de la constante π est stockée dans la variable `np.pi`.

Le module numpy.random.

- Exemple d'importation : `import numpy. random as rd`.
- La commande `rd.normal()` renvoie une réalisation aléatoire de la loi normale centrée réduite.

Annexe B - Commandes SQL

La fonction COUNT(). La fonction d'agrégation `COUNT()` permet de connaître le nombre d'enregistrements d'une table, vérifiant éventuellement une certaine condition.
Nous donnons ci-dessous plusieurs exemples d'utilisation de la fonction `COUNT()`, en considérant une table nommée `ma_table` comportant deux colonnes `colonne_1` et `colonne_2`.

- La requête suivante renvoie le nombre total d'enregistrements dans `ma_table` :

```
SELECT COUNT(*) FROM ma_table
```

- La requête suivante renvoie le nombre d'enregistrements de `ma_table` vérifiant la condition `cond` :

```
SELECT COUNT(*) FROM ma_table WHERE cond
```

- La requête suivante renvoie le nombre d'enregistrements de `ma_table` pour lesquels la valeur de `colonne_2` n'est pas vide :

```
SELECT COUNT(colonne_2) FROM ma_table
```

La fonction d'agrégation AVG(). La fonction `AVG()` permet de calculer la moyenne des valeurs d'une colonne dans une table.
Par exemple, si on considère la table nommée `table` contenant les enregistrements suivants :

colonne_1	colonne_2	colonne_3	colonne_4
1	69	Lyon	4
2	31	Toulouse	8
3	54	Nancy	5
4	64	Saint-Jean-de-Luz	17
5	44	Nantes	6

alors la requête suivante

```
SELECT AVG(colonne_4) FROM table WHERE colonne_1<=3
```

affiche la moyenne des valeurs de `colonne_4` des trois premiers enregistrements : 5.6667 c'est-à-dire $\dfrac{4+8+5}{3}$.

★★★★★★★ Fin ★★★★★★★