DS7

3h00

- 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 amené·e ·s 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.

Exercice 1. On considère deux urnes U et V. L'urne U contient 2 jetons rouges et 4 jetons noirs. L'urne V contient 3 jetons rouges et 3 jetons noirs. On effectue une succession de tirages avec remise dans ces deux urnes selon le protocole suivant :

— On choisit une urne au hasard au départ et on tire un jeton.

— A chaque fois que l'on tire un jeton rouge, on change d'urne pour effectuer le tirage suivant. En revanche, si on tire un jeton noir, on reste dans la même urne pour le tirage suivant.

Pour tout $n \in \mathbb{N}^\star$, on note : U_n (respectivement V_n) l'événement le n-ième tirage s'effectue dans l'urne U (respectivement le n-ième tirage s'effectue dans l'urne V), R_n l'événement : le n-ième jeton tiré est rouge. Enfin, on pose : $p_n = P(U_n)$ et $q_n = P(V_n)$.

1. Informatique
   (a) Quelle bibliothèque Python permet de simuler des expériences aléatoires ?
   (b) Créer deux listes `UrneU` et `UrneV` constituées des caractères : `'R'` et `'N'` représentant les deux compositions d'urnes.
   (c) Définir une fonction `tirage(L)` prenant comme argument une liste `L` dont chaque élément est un caractère 'R' ou 'N' et qui retourne au hasard un élément de cette liste.
   (d) Définir une fonction `tirage_U(nom)` prenant comme argument une chaîne de caractères `nom` simulant un tirage dans l'urne U si `nom=='U'` et dans V si `nom=='V'`
   (e) Compléter (en recopiant sur votre copie) le script suivant, de sorte qu'il renvoie la liste des n premiers résultats de tirages effectués selon la règle donnée précédemment.

```
def experience(n):
    p=.........
    if p==0:
        urne='U'
    else:
        urne='V'
    resultat=tirag_U(urne)
    liste=[resultat]
    for i in range(n-1):
        if resultat=='N':
            resultat=......
        else:
            if urne='V'
                urne=....

            else:
                ......
            resultat=.....
        liste=.......
    return ............
```

2. Calculer p_1, q_1, p_2 et q_2.
3. Démontrer que, pour tout $n \geq 2$, on a : $p_n = \frac{2}{3}p_{n-1} + \frac{1}{2}q_{n-1}$.
4. En déduire que, pour tout $n \in \mathbb{N}^\star$, on a : $p_n = \frac{3}{5} - \frac{1}{10}\left(\frac{1}{6}\right)^{n-1}$, puis la valeur de q_n.
5. sont les limites des suites $(p_n)_{n\in\mathbb{N}^\star}$ et $(q_n)_{n\in\mathbb{N}^\star}$ quand n tend vers l'infini ? Est-ce cohérent ?
6. On tire un jeton rouge au n-ième tirage ($n \in \mathbb{N}^\star$). Quelle est la probabilité que ce jeton provienne de l'urne U ?

   Si on note r_n cette probabilité, quelle est la limite de la suite $(r_n)_{n\in\mathbb{N}^\star}$ quand n tend vers l'infini ?

Exercice 2. Soit f la fonction définie par

$$f(x) = \exp(\frac{-1}{x^2})$$

1. Justifier que f est dérivable sur $\mathbb{R}^*$ et calculer sa dérivée.
2. Montrer que f est prolongeable par continuité en 0. On note encore f la fonction ainsi prolongée.
3. Montrer que f est dérivable en 0 (on étudiera le taux d'accroissement)
4. La fonction f' est-elle continue sur $\mathbb{R}$?
5. On admet que pour tout n il existe un polynôme $P_n \in \mathbb{R}[X]$ tel que pour tout $x \in \mathbb{R}^*$

$$f^{(n)}(x) = P_n(\frac{1}{x})f(x)$$

. (Ici $f^{(n)}$ désigne la dérivée $n-$ éme de f)

   (a) Expliciter les polynômes P_0 et P_1

   (b) Montrer que pour tout $n \in \mathbb{N}$ et tout $x \in \mathbb{R}^*$:

$$P_{n+1}(\frac{1}{x}) = -\frac{1}{x^2}P_n'(\frac{1}{x}) + 2\frac{1}{x^3}P_n(\frac{1}{x})$$

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

$$P_{n+1} = -X^2P_n' + 2X^3P_n$$

   (d) Soit d_n le degré de P_n, justifier que $d_{n+1} = d_n + 3$ et en déduire la valeur de d_n.

Exercice 3. Ce problème propose d'étudier quelques propriétés des polynômes palindromiques, c'est-à-dire dont les coefficients peuvent être indifféremment lus dans l'ordre croissant ou décroissant des degrés. Par exemple, le polynôme $2X^6 - X^4 + 5X^3 - X^2 + 2$ est palindromique.

1. Cas du degré 2. Soit $P_2 = aX^2 + bX + a \in \mathbb{C}[X]$ un polynôme palindromique de degré 2 .
   (a) Justifier que 0 n'est pas racine de P_2.
   (b) Rappeler le lien entre les valeurs des racines r_1, r_2 d'un polynôme $\alpha X^2 + \beta X + \gamma$ et ses coefficients (α, β, γ) . En déduire que si P_2 n'admet pas de racine double alors le produit de ses racines vaut 1 et aucune de ses racines n'est égale à 1 ou -1 .
   (c) Montrer que si P_2 admet une racine double alors cette racine vaut 1 ou -1 .
2. Cas du degré 3. Soit $P_3 = aX^3 + bX^2 + bX + a \in \mathbb{C}[X]$ un polynôme palindromique de degré 3.
   (a) Trouver une racine évidente de P_3.
   (b) Soit $Q_3 \in \mathbb{C}[X]$ tel que $P_3 = (X + 1)Q_3$. Montrer que Q_3 est palindromique de degré 2 .
   (c) Que peut-on déduire des résultats précédents sur la multiplicité des racines de P_3 ? (on étudiera différents cas selon la multiplicité des racines de Q_3)
3. Un exemple de degré 4. Soit $P_4 = X^4 - 5X^3 + 6X^2 - 5X + 1$.
   (a) Développer l'expression $(\alpha + 1/\alpha)^2$ pour tout $\alpha \in \mathbb{C}^\star$.
   (b) Montrer que $\alpha \in \mathbb{C}$ est racine de P_4 si et seulement si $\alpha+1/\alpha$ est racine de $Q_4 = X^2-5X+4$
   (c) En déduire la factorisation de P_4 dans $\mathbb{C}[X]$.
4. Cas général. Soit $P = \sum_{k=0}^{d} a_k X^k \in \mathbb{C}[X]$ un polynôme palindromique de degré $d \geqslant 0$. On remarque que les coefficients de P vérifient $a_{d-k} = a_k$ pour tout $k \in [\![0, d]\!]$.
   (a) Justifier que 0 n'est pas racine de P.
   (b) On suppose que le degré de d est impair, on l'écrit $d = 2N + 1$ avec $N \in \mathbb{N}$. Montrer que

$$P = \sum_{k=0}^{N} a_k(X^k + X^{2N+1-k})$$

   (c) En déduire que si d est impair alors -1 est racine de P.

(d) Montrer que si $\alpha \in \mathbb{C}$ est racine de P alors $1/\alpha$ aussi (on ne suppose pas ici que d est impair).

5. (INFORMATIQUE) On représente informatiquement un polynôme sous forme d'une liste de ses coefficients ie. `[a0,a1, ... ,an]` représente le polynôme $\sum_{k=0}^{n} a_k X^k$.

(a) Quelle polynôme représente la liste [2,3,4,0,0] ?

(b) Ecrire une fonction Python `simplifie(P)` qui prend en argument une liste `P` correspondant à un polynôme P et retourne la liste qui représente le polynome P mais où le dernier terme est non nul. Exemple = `simplifie( [2,3,4,0,0])` retournera [2,3,4]

(c) Completer (en recopiant sur votre copie) le code Python `est_palyndrome(P)` qui prend en argument une liste `P` correspondant à un polynôme P et retourne `True` si le polynôme est palyndromique et `False` sinon.

```
def est_palyndrome(P):
    P=simplifie(P)
    for k in range(len(P)):
        if P[k]!=P[......]:
            return .......
    return ......
```

(d) Ecrire une fonction Python `evaluation(P,x)` qui prend en argument une liste `P` correspondant à un polynôme P et un flottant `x` et retourne la valeur de $P(x)$

(e) On souhaite coder une fonction Python `derivation(P)` qui prend en argument une liste `P` correspondant à un polynôme P et retourne la liste des coefficients correspondant au polynôme P'. Exemple : Si $P = 2 + 3X + 4X^2$ le polynôme dérivé est $P' = 3 + 8X$, donc `derivation([2,3,4])` retournera `[3,8]`. Des codes suivants lequel correspond à cette procédure. On explicitera ce que rendent les autres codes (erreurs ou valeur de la liste avec l'exemple [2,3,4])

```
def derivation1(P):
    D=[]
    for i in range(1,len(P)):
        D=D+[P[i]*i]
    return  D
```

```
def derivation2(P):
    D=[]
    for i in range(len(P)):
        D=D+[P[i]*i]
    return  D
```

```
def derivation3(P):
    D=[]
    for i in range(len(P)):
        D=D+[P[i+1]*i]
    return  D
```

```
def derivation4(P):
    D=[]
    for i in range(1,len(P)):
        D=D+[P[i]*(i-1)]
    return  D
```

(f) Ecrire une fonction `multiplicite(P,x)` qui prend en argument une liste `P` correspondant à un polynôme P et un flottant `x` et retourne la multiplicité de x (si x n'est pas racine de P, la fonction retournera 0, si x est racine de multiplicité 1 de P, la fonction retournera 1, etc...)