DS 5

Durée 3h00

- Les calculatrices sont <u>interdites</u> 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.

Exercice 1. On condière un ensemble de personnes composé de n_1 hommes et de n_2 femmes. On désire élire un bureau de p représentants choisis parmi ces (n_1+n_2) personnes.

1. Combien y-a-t-il de bureaux possibles ?
2. Combien y-a-t-il de bureaux possibles contenant exactement k hommes.
3. En déduire la relation suivante : Pour tout $(n_1, n_2, p) \in \mathbb{N}^3$, tel que $p \leq n_1$ et $p \leq n_2$:

$$\binom{n_1+n_2}{p} = \sum_{k=0}^{p} \binom{n_1}{k}\binom{n_2}{p-k}$$

Exercice 2. On considère l'équation suivante , d'inconnue $z \in \mathbb{C}$:

$$z^3 + z + 1 = 0 \qquad (E)$$

1. On note $f : \mathbb{R} \mapsto \mathbb{R}$, la fonciton définie par $f(t) = t^3 + t + 1$. A l'aide de l'étude de f, justifier que l'équation (E) possède une unique solution réelle, que l'on notera r. Montrer que $r \in]-1, \frac{-1}{2}[$.
2. On note z_1 et z_2 les deux autres solutions complexes de (E) qu'on ne cherche pas à calculer. On sait alors que le polynôme $P(X) = X^3+X+1$ se factorise de la manière suivante :
$$P(X) = (X-r)(X-z_1)(X-z_2).$$
En déduire que $z_1 + z_2 = -r$ et $z_1 z_2 = \frac{-1}{r}$.
3. Justifier l'encadrement : $\frac{1}{2} < |z_1 + z_2| < 1$.
De même montrer que $1 < |z_1 z_2| < 2$.
4. Rappeler l'inégalité triangulaire. En déduire que pour tout $x, y \in \mathbb{C}$
$$|x - y| \geq |x| - |y|$$
5. Montrer alors que
$$|z_1 + z_2| > |z_1| - \frac{2}{|z_1|}$$
6. Grâce à un raisonnement par l'absurde montrer que $|z_1| < 2$.
7. Conclure que toutes les solutions de (E) sont de modules strictement inférieures à 2.

Exercice 3. Dans l'espace $\mathcal{E}$ muni d'un repère orthonormé, on considère la droite $\mathcal{D}$ de représenation paramétrique :

$$\mathcal{D} : \begin{cases} x &= -3+t \\ y &= 1+2t \\ z &= 2-t \end{cases}, t \in \mathbb{R}$$

Pour tout ce problème on fixe un point $A(\alpha, \beta, \gamma)$ et on note $H(\lambda, \mu, \nu)$ son projeté orthogonal sur $\mathcal{D}$.

1. Soit $\mathcal{P}_1$ le plan contenant les points $B(-1,0,0), C(2,7,-3)$ et $D(-2,4,1)$.
   (a) Montrer que $\mathcal{P}_1$ a pour équation $x + z + 1 = 0$
   (b) Montrer que $\mathcal{P}_1$ contient $\mathcal{D}$.
2. Soit $\mathcal{P}_2$ le plan contenant $\mathcal{D}$ et le point $E(-2,3,-1)$.

(a) Donner deux vecteurs parallèles à $\mathcal{P}_2$ qui ne sont pas colinéaires entre eux.

(b) En déduire un vecteur orthogonal à $\mathcal{P}_2$.

(c) Montrer alors que $-2x + y - 7 = 0$ est une équation cartésienne de $\mathcal{P}_2$

3. Soit $\mathcal{P}_3$ le plan perpendiculaire à $\mathcal{D}$ et passant par A.

   (a) Donner un vecteur directeur de $\mathcal{D}$.

   (b) Déterminer une équation cartésienne de $\mathcal{P}_3$. (En fonction évidemment de α, β, γ)

4. En déduire que les coordonnées de H vérifient un système linéaire qu'on peut écrire sous la forme :

$$M_1 \begin{pmatrix} \lambda \\ \mu \\ \nu \end{pmatrix} = M_2 \quad \text{où } M_1 = \begin{pmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \\ 1 & 2 & -1 \end{pmatrix}$$

   et $M_2 \in \mathcal{M}_{3,1}(\mathbb{R})$ est une matrice colonne à déterminer (qui dépendra de (α, β, γ)

5. Montrer que M_1 est inversible et calculer son inverse.

6. En déduire que les coordonnées de H sont données par :

$$\begin{pmatrix} \lambda \\ \mu \\ \nu \end{pmatrix} = P_1 \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} + P_2 \quad \text{où } P_1 = \frac{1}{6} \begin{pmatrix} 1 & 2 & -1 \\ 2 & 4 & -2 \\ -1 & -2 & 1 \end{pmatrix} \text{ et } P_2 = \frac{1}{2} \begin{pmatrix} -5 \\ 4 \\ 3 \end{pmatrix}$$

7. (a) Déterminer en fonction de α, β et γ la valeur du paramètre $t \in \mathbb{R}$ du point $M(x, y, z) \in \mathcal{D}$ telle que le veteur $\vec{AM}$ soit orthogonal au vecteur $\vec{u} = (1, 2, -1)$

   (b) Retrouver le résultat de la question 6 à l'aide de la valeur du paramétre t obtenue à la question précédente.

Exercice 4. On souhaite dans cet exercice coder le jeu du master mind. On en rappelle les règles briévement. Le jeu se joue à deux : un codificateur et un décodeur. Le but est de deviner, par déductions successives, la couleur et la position des 4 pions cachés derrière un écran. Déroulement du jeu : le codificateur (qui dans notre cas sera joué par l'ordinateur) crée un code avec 4 pions de 5 couleurs différentes. Il doit prendre soin de ne pas révéler la couleur et la répartition dans les trous des pions. Il n'y a pas de restrictions sur le choix des différents pions.

Son adversaire, le décodeur (vous), est chargé de déchiffrer ce code secret. Il doit le faire en 10 coups au plus. Il place 4 pions dans les trous de la première rangée immédiatement près de lui. Si l'un des pions correspond par sa position et sa couleur à un pion caché derrière l'écran, le codificateur l'indique en plaçant une fiche noire dans l'un des trous de marque, sur le côté droit correspondant du plateau. Si l'un des pions correspond uniquement par sa couleur, le Codificateur l'indique par une fiche blanche dans l'un des trous de marque. S'il n'y a aucune correspondance, il ne marque rien.

On propose de coder les 5 couleurs par leur premières lettres : Jaune : 'J', Rouge :'R', Marron : 'M', Bleu : 'B' et vert 'V'. On appelle `couleur` la liste définie par `couleur = ['J','R','M','B','V']`. (On pourra l'utiliser telle quelle dans l'écriture des programmes).

Exemple : Supposons que le code à deviner soit `['J','M','R','M']`. et que le code proposé soit `['J','M','M','V']`.

— On s'intéresse à la couleur jaune 'J'. Elle est en position `[0]` dans le code à deviner et `[0]` dans le code proposé. Il y aura donc une fiche noire.
— On s'intéresse à la couleur marron 'M'. Elle est en position `[1,3]` dans le code à deviner et `[1,2]` dans le code proposé. Il y aura donc une fiche noire et une fiche blanche
— On s'intéresse à la couleur rouge 'R'. Elle est en position `[2]` dans le code à deviner et elle n'est pas dans le code proposé. Il y aura donc aucune fiche.
— On s'intéresse à la couleur vert 'V'. Elle n'est pas dans le code à deviner et en position `[3]` dans le code proposé. Il y aura donc aucune fiche
— On s'intéresse à la couleur bleu 'B'. Elle n'est ni dans le code cherché ni dans le code proposé, il n'y aura aucune fiche.

Au final l'ordinateur devra afficher 2 fiches noires et 1 fiche blanche.

1. (a) Combien y-a-t-il de codes possibles au MasterMind ?
   (b) Combien y-a-t-il de codes possibles au MasterMind avec au plus 1 rouge ?
   (c) Combien y-a-t-il de codes possibles au MasterMind avec exactement 1 rouge ?
   (d) Combien y-a-t-il de codes possibles au MasterMind où toutes les couleurs sont différentes ?
2. Ecrire une fonction `code` qui retourne un code aléatoire (sous forme de liste) valable au mastermind.
3. (a) Ecrire une fonction `place` qui prend en argument une liste `L` (qui correspond à un code) et une lettre `a` (qui correspond à une couleur) et retourne la liste des positions de la lettre `a` dans la liste `L`.
   (b) Que retourne `place(['J','R','V','R'],'R')` ?
   (c) Que retourne `place(['J','R','V','R'],'M')` ?
4. On va tout d'abord essayer de comparer les deux codes (celui cherché et celui proposé par le joueur) pour une couleur donnée.
   Compléter la fonction `compare_deux_couleur` qui prend en argument deux listes `L1` et `L2` qui correspond à la position des jetons d'une couleur donnée dans le code et dans la proposition faite par le joueur et retourne deux nombres (b,n) correspondant au nombre de fiches blanches et noires que le codificateur doit donner comme réponse.
   Exemples :
   — compare_deux_couleur([1,4],[1,2]) retourne (1,1), c'est-à-dire 1 blanche et 1 noire. La noire correspond à la couleur bien placée en position 1 et la blanche correspond à la couleur mal placée en position 2 (qui est en position 4 dans le code)
   — compare_deux_couleur([1],[1,2]) retourne (0,1), c'est-à-dire 0 blanche et 1 noire. La noire correspond à la couleur bien placée en position 1 .
   — compare_deux_couleur([],[1,2]) retourne (0,0), c'est-à-dire 0 blanche, 0 noire. Il n'y a pas la couleur correspondante dans le code.
   — compare_deux_couleur([3],[1,2]) retourne (1,0), c'est-à-dire 1 blanche, 0 noire. Une couleur est mal placée l'autre est en trop.
5. (a) Ecrire une fonction `couleur_distincte` qui prend en argument une liste `L` (qui correspond à un code) et qui retourne la liste des couleurs distinctes dans `L`.

(b) Que retourne `couleur_distincte(['J','J','V','R'])` ?

6. Ecrire une fonction `decode` qui prend en argument deux listes `code` et `proposition` (qui correspondent respectivement au code cherché et au code proposé par le joueur) et qui retourne une liste correspondant à la réponse que doit faire le Codificateur : la liste contiendra donc un certain nombre de fiches noires (codée par la lettre 'N') correspondant au nombre de pions bien placés dans la liste proposée par le joueur et un certain nombre de fiches blanches.
7. Ecrire une fonction `transform` qui prend en argument une chaine de caractères `S` et retourne une liste dont chaque entrée est une lettre de la chaine `S`.
8. Compléter la fonction suivante qui permet de jouer au master mind : (on s'arretera si le joueur gagne ou si il dépasse le nombre de coups autorisés)

```
def master_mind():
    code_cherche=
    c=0 #compteur qui permet de compter le nombre de
        #coups joues par le joueur
    proposition = [] # On initialise avec une proposition vide

    #tant que le joueur n'a pas depasse le nombre de propositions
    #et qu'il n'a pas trouve le code :
    while
       #le joueur inscrit un code sous forme de  chaine de caracteres:
        prop = input('quel est le code ?')

        #que l'on transforme en liste :
        proposition = transform(prop)

       #on calcule le nombre de fiches noires et blanches
        B,N =
        print('Fiches blanches:', B, 'Fiches noires:', N)

        #et on incremente le compteur
        c =

    if code_cherche ==              :
        print
    else:
        print
```