Feuille Oraux_5 : Du cours, un exercice à préparer et 10mn sans préparation.

Du cours. *(Six élèves aux tableaux : une suite de 4 questions. (10 mn))*

Tableau 1 : 1) Définition de : "La suite de variables aléatoires $(X_n)_{n\in\mathbb{N}}$ converge en loi vers une variable aléatoire X à densité".
2) Donner le développement limité d'ordre 4 de e^x en 0.
3) Définition de la matrice d'une application linéaire de $\mathbb{K}^p$ dans $\mathbb{K}^n$ dans des bases.
4) Donner les dérivées $k-$ième du monôme X^n.

Tableau 2 : 1) Donner le théorème central limite.
2) Donner le théorème des sommes de Riemann.
3) Définition du minimum d'une partie de $\mathbb{R}$ et d'un minorant d'une partie non vide de $\mathbb{R}$.
4) Enoncer le théorème des valeurs intermédiaires.

Tableau 3 : 1) Donner la densité continue de la loi normale centrée réduite et l'allure de sa représentation graphique.
2) Enoncer le théorème donnant l'expression de (u_n) vérifiant $\forall n \in \mathbb{N},\ u_{n+2} = au_{n+1} + bu_n$
3) Pour $(A_n)_{n\in\mathbb{N}}$ une suite d'événements, donner à la définition de $\bigcap_{n=0}^{+\infty} A_n$ et $\bigcup_{n=0}^{+\infty} A_n$
4) Théorème de la limite monotone pour une suite réelle.

Tableau 4 : 1) Donner le théorème de d'Alembert-Gauss.
2) Enoncer le théorème dit "des gendarmes" pour les suites.
3) Donner la dérivée de $t \longmapsto f(x(t), y(t))$
4) Donner la somme des n premiers entiers et des n premiers carrés.

Tableau 5 : 1) Donner la définition de la loi binomiale de paramètre (n, p), son espérance et sa variance.
2) Définition d'un système complet d'événements (version dénombrable).
3) Caractérisation de la projection orthogonale sur un sous-espace vectoriel F de $\mathbb{R}^n$.
4) Définition du coefficient binomial $\binom{n}{k}$ pour k et n deux entiers.

Tableau 6 : 1) Donner la définition de la loi de Poisson de paramètre λ, son espérance et sa variance.
2) Enoncer le théorème donnant la dérivée de $g \circ f$ lorsque f et g sont deux fonctions dérivables.
3) Définition de la matrice d'un endomorphisme dans une base.
4) Définition d'une bijection.

D'autres questions de cours :

Un exercice court *(Six élèves au tableau après 10 minutes de préparation.)*

Tableau 1 : Soient x et y deux réels, on note M la matrice $\begin{pmatrix} x & y \\ y & x \end{pmatrix}$.

1) Donner une condition nécessaire et suffisante sur y et x pour que cette matrice soit inversible.
2) Soit $(\Omega, \mathcal{A}, P)$ un espace probabilisé . On considère deux variables aléatoires indépendantes X et Y définies sur cet espace probabilisé, indépendantes, et suivant une loi géométrique de paramètre $p \in]0,1[$.

   Pour tout ω de Ω on définit la matrice $M(\omega) = \begin{pmatrix} X(\omega) & Y(\omega) \\ Y(\omega) & X(\omega) \end{pmatrix}$.

   Quelle est la probabilité que cette matrice soit inversible ?

Tableau 2 : Soient X et Y deux variables aléatoires indépendantes de même loi normale centrée réduite.

On pose $Z = \sup(X, Y)$.

1) Montrer que Z est à densité et donner sa fonction de densité g en fonction de la densité f et de la fonction de répartition Φ communes à X et Y.
2) Montrer que Z admet une espérance et la déterminer à l'aide d'une intégration par parties.

Tableau 3 : 1) Diagonaliser si possible les matrices :

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} -4 & 3 \\ -6 & 5 \end{pmatrix}$$

2) Donner, M^n, $n \in \mathbb{N}$, la puissance $n$ième de la matrice

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -4 & 3 \\ 0 & 0 & -6 & 5 \end{pmatrix}$$

Tableau 4 : On considère l'équation différentielle :

$$(E) : \quad y'' - 2y' + 2y = 2x^2$$

Résoudre (E) et déterminer l'unique solution f de (E) telle que $f(0) = 1$ et $f'(0) = 3$.

Indication : On cherchera une solution particulière polynomiale.

Tableau 5 : Soit $u = (u_1, ..., u_n)$ un vecteur unitaire *(de norme 1)* et f l'endomorphisme de $\mathbb{R}^n$ canoniquement associé à la matrice A de $\mathscr{M}_n(\mathbb{R})$ ayant pour coefficients $u_i u_j$, où $(i,j) \in [\![1,n]\!]^2$.

1) Montrer que f est diagonalisable.
2) Montrer que $\mathrm{Im}(f) = \mathrm{Vect}(u)$.
3) Soit $x \in \mathbb{R}^n$.
   a. Calculer $f(x) - \langle u|x\rangle u$. *($\langle u|x\rangle$ est le produit scalaire de u et x).*
   b. Quelle est la nature de f.
   c. En déduire le spectre de f.

Tableau 6 : Lorsque X est une variable aléatoire discrète prenant des valeurs dans $\mathbb{N}$,

on définit la fonction génératrice de X $g_X : [0,1] \to \mathbb{R}$ par :

$$\forall t \in [0,1], \quad g_X(t) = \sum_{k=0}^{+\infty} P(X = k)\, t^k.$$

1) Justifier que g_X est bien définie sur $[0,1]$, puis calculer $g_X(1)$.
2) Déterminer la fonction génératrice d'une variable aléatoire suivant une loi géométrique.

Jusqu'à la fin de l'exercice X est une variable finie à valeurs dans $[\![0,n]\!]$.

3) Prouver que g_X est de classe C^∞ sur $[0,1]$, exprimer l'espérance de X en onction de g'_X puis donner une relation entre $V(X), g''_X(1), g'_X(1)$.
4) Déterminer la fonction génératrice de X lorsque X suit une loi binomiale. Retrouver alors les formules d'espérance et de variance d'une variable binomiale.

Un exercice d'informatique sans préparation. *(six élèves au tableau, les échanges seront limités)*

Tableau 1 :

1) Donner une fonction `indices_max` qui prend en entrée une liste de nombres et qui renvoie la liste des indices où le maximum de la liste est atteint.
2) Donner une fonction `compter_occurrences` qui prend en entrée une liste de nombres entiers compris entre 0 et $n-1$, et qui renvoie une liste des nombres d'occurrences de chaque entier de 0 à $n-1$.
3) Donner une fonction `valeurs_modales` qui prend en entrée une liste de nombres entiers entre 0 et $n-1$ et qui renvoie la liste des valeurs modales (c'est-à-dire les entiers qui apparaissent le plus souvent).

Tableau 2 :

1) Écrire une fonction `premier_indice` qui prend en entrée une liste de nombres et une valeur, et qui renvoie le premier indice où cette valeur apparaît dans la liste. Si la valeur n'est pas trouvée, la fonction doit renvoyer `-1`.
2) Écrire une fonction `inverse_liste` qui prend en entrée une liste et renvoie une nouvelle liste contenant les mêmes éléments dans l'ordre inverse.
3) Écrire une fonction `sous_liste` qui prend en entrée une liste et deux indices `début` et `fin`, et qui inverse l'ordre des éléments compris entre ces indices (inclus au début, exclu à la fin). Si les indices sont invalides, la fonction ne fait rien.

Tableau 3 :

1) Écrire une fonction `extraire_annee` qui prend en entrée une liste de dates au format `"JJ/MM/AAAA"` *(chaînes de caractères)* et qui renvoie une liste des années extraites *(entiers)*.
2) En utilisant `extraire_annee`, écrire une fonction `compter_annees` qui prend en entrée une liste de dates `Date` et qui renvoie deux listes : la liste des années et la liste du nombre de fois où elle apparait dans `Date`.
3) En utilisant `compter_annees`, écrire une fonction `annee_plus_fréquente` qui prend en entrée une liste de dates et qui renvoie la ou les années les plus fréquentes *(sous forme de liste)*.

Tableau 4 : Une image est codée par un tableau `numpy` contenant des entiers de 0 à 255.

1) Ecrire une fonction qui prend en entrée un tableau `numpy` carré à deux dimensions et qui teste que ce tableau contient bien des entiers compris entre 0 et 255.
2) Écrire une fonction qui prend en entrée un tableau `numpy` carré à deux dimensions et qui renvoie une nouvelle image où chaque pixel est remplacé par la moyenne des pixels dans sa zone 3x3 centrée sur lui. On pourra *considérer que les pixels sur les bords conservent leur valeur d'origine.*
3) Écrire une fonction qui prend en entrée un tableau `numpy` carré à deux dimensions et qui effectue un miroir horizontal de l'image *(symétrie par rapport à un axe vertical central)*.

Tableau 5 :

1) Écrire une fonction `est_email_valide(email)` qui prend en paramètre une chaîne de caractères `email` et qui renvoie `True` si cette chaîne respecte la forme simplifiée suivante : contient exactement un symbole `@`, au moins un caractère avant et après ce symbole, et au moins un point `.` dans la partie après le `@`. Sinon, la fonction renvoie `False`.
2) Écrire une fonction `extraire_domaine(email)` qui prend en paramètre une chaîne `email` supposée valide, et qui renvoie la partie après le symbole `@` (le domaine).
3) Écrire une fonction `filtrer_emails_domaine(liste_emails, domaine)` qui prend en paramètres une liste de chaînes `liste_emails` et une chaîne `domaine`, et qui renvoie la liste des emails de `liste_emails` dont le domaine correspond exactement à `domaine`.

Tableau 6 : Une grille carrée représente un **terrain de jeu** où chaque case contient un entier correspondant à une **altitude**. Par exemple, la grille suivante (codée par un tableau `numpy`) : $\begin{bmatrix} 2 & 3 & 4 & 2 \\ 5 & 1 & 3 & 4 \\ 2 & 3 & 2 & 1 \\ 4 & 2 & 1 & 3 \end{bmatrix}$ correspond à un terrain où chaque case a une altitude donnée.

1) Écrire une fonction `voisins(i, j, n)` qui renvoie la liste des positions des voisins valides de la case (i, j) dans une grille carrée de taille $n \times n$ *(au maximum 8 voisins)*.
2) Écrire une fonction qui prend en argument une grille d'altitudes et renvoie la liste des positions (i, j) des *pics locaux*, c'est-à-dire des cases dont l'altitude est strictement supérieure à celle de toutes leurs cases voisines.
3) Écrire une fonction qui prend en argument une grille d'altitudes et renvoie une nouvelle grille où chaque case contient la différence d'altitude entre la case et la moyenne des altitudes de ses voisins *(en considérant 0 comme altitude des voisins hors grille)*.