Informatique
Préparation à l'oral.

Algèbre et arithmétique - Sujets

2016

1. D'après Centrale (PSI) ♠♠

A tout polynôme P on associe $S(P)=\sum_{k=0}^{+\infty}\frac{P(k)}{k!}$.

a) Justifier l'existence de $S(P)$. Montrer que S est une forme linéaire.

b) Avec Python, calculer $\sum_{k=0}^{50}\frac{P(k)}{k!}$ pour $P=X^d$ avec $d\in[\![0,10]\!]$, puis pour un polynôme de degré 9 de votre choix.

c) On pose $H_0=1$, puis, pour tout entier naturel n, $H_{n+1}=(X-n)H_n$. Montrer que $(H_k)_{0\le k\le n}$ est une base de $\mathbb{R}_n[X]$.

d) Calculer $S(H_n)$ pour tout entier naturel n. En déduire une méthode pour calculer $S(P)$ pour P quelconque.

e) Avec Python, écrire un programme permettant de calculer les coefficients de H_n.

Questions complémentaires :

f) A l'aide de la fonction de la question précédente, écrire une fonction Python `MatPassageHn` construisant un tableau `numpy` correspondant à la matrice de passage de la base canonique de $\mathbb{R}_n[X]$ à la base $(H_k)_{0\le k\le n}$. La fonction `MatPassageHn` recevra comme seul argument l'entier naturel n.

g) Mettre en œuvre la méthode proposée à la question d) et comparer avec les résultats obtenus à la question a).

2. Centrale (PSI) ♠♠

Soit C_n la matrice de $\mathscr{M}_n(\mathbb{R})$ telle que $c_{i,j}=1$ si $i\in\{1,\ldots,n-1\}$ et $j=i+1$ et si $(i,\,j)=(n,1)$, et $c_{i,j}=0$ sinon.

a) *i*) Ecrire la matrice C_6.

ii) Ecrire un programme donnant C_n pour tout $n\geq 1$.

iii) Vérifier que $C_n^n=I_n$ pour $1\leq n\leq 10$.

b) *i*) Prouver ce résultat pour tout *n*.

ii) En déduire que C_n est diagonalisable sur $\mathbb{C}$. Est-elle diagonalisable sur $\mathbb{R}$?

iii) Montrer que 1 est valeur propre de C_n. Déterminer toutes les valeurs propres de C_n.

3. Centrale (PSI) ♠♠

On considère deux matrices M_n et T_n dans $\mathscr{M}_n(\mathbb{R})$ avec $(M_n)_{i,j}=\min(i,\,j)$ et T_n triangulaire supérieure avec des 1 sur le triangle supérieur y compris la diagonale.

a) Ecrire un script Python permettant l'affichage de M_n pour $2\leq n\leq 10$.

b) Calculer $\det(M_n)$ pour $n=2,\ldots,10$. Conjecturer un résultat.

c) Ecrire un script Python permettant l'affichage de T_n pour $2\leq n\leq 10$.

d) Pour $n=2,\ldots,10$, calculer ${}^tT_nT_n$. Conjecturer un résultat.

e) Démontrer les deux conjectures faites.

f) Montrer que M_n est diagonalisable à valeurs propres strictement positives.

g) Montrer que $\lambda_n=\max(\mathrm{Sp}(M_n))\geq\dfrac{n+1}{2}$.

h) En essayant de minimiser les calculs, trouver M_n^{-1}.

i) Utiliser Python pour le calcul de M_n^{-1}.

2015

1. D'après ENSAM 2015 (PSI) ♠♠

En Python :

1. A la console, écrire une instruction permettant de créer la liste L30 contenant 31 `True`.
2. Ecrire une fonction « modifier » recevant en argument une liste L (ne contenant que des éléments valant `True` ou `False`) et un entier naturel *p* et telle que :
    a. Si $p = 0$, le premier terme de L est positionné à `False`.
    b. Si $p = 1$, le second terme de L est positionné à `False`.
    c. Si $p \notin \{0;1\}$, le *k*-ième élément de L est positionné à `False` dès que $k \neq p$ est un multiple de *p*.

   Tester la fonction avec L30.
3. Ecrire une fonction « premiers » recevant en argument un entier naturel *n* et renvoyant la liste des entiers naturels premiers inférieurs ou égaux à *n*.
4. 823417 est-il premier ?
5. Quel est l'algorithme mis en œuvre dans cet exercice ?

2. ENSAM 2015 (PSI) ♠♠

Les diviseurs propres d'un entier naturel sont les entiers naturels qui lui sont strictement inférieurs et le divisent. Par exemple, les diviseurs propres de 100 sont : 1, 2, 4, 5, 10, 20, 25 et 50.

En Python :

1. Ecrire une fonction « LDP » recevant en argument un entier naturel *n* non nul et renvoyant une liste contenant les diviseurs propres de *n* (dans l'ordre croissant).
2. Ecrire une fonction « SDP » recevant en argument un entier naturel n non nul et renvoyant la somme de ses diviseurs propres (pour $n = 100$, la fonction renverra 117).
3. Un entier naturel non nul est dit « parfait » s'il est égal à la somme de ses diviseurs propres. Ecrire une fonction « Parfaits » recevant en argument un entier naturel K non nul et renvoyant la liste (ordonnée dans l'ordre croissant) des entiers naturels non nuls parfaits inférieurs ou égaux à K.

4. Deux entiers naturels non nuls sont dits « Amicaux » lorsque chacun est égal à la somme des diviseurs de l'autre. Ecrire une fonction « Amicaux » recevant comme argument un entier naturel supérieur ou égal à 2 et renvoyant tous les couples d'entiers amicaux (p, q) vérifiant $p < q \leq \mathrm{K}$.

3. ENSAM 2015 (PSI) ♠♠

En Python :

1. Ecrire une fonction qui génère deux matrices carrées A et B de dimension respective 9 et 2 et dont tous les coefficients sont des entiers aléatoires de $[\![0\,;9]\!]$.
2. Ecrire une fonction qui retourne *True* ou *False* selon qu'une matrice *B* est ou non un motif d'une matrice *A*, les matrices *A* et *B* étant de taille quelconque.
3. Ecrire une fonction retournant le plus petit coefficient d'une matrice A.
4. Ecrire une fonction retournant $S = \sum_{i\in[\![1\,;\,n]\!],\ j\in[\![1\,;\,p]\!]} \left(a_{ij} - b_{ij}\right)^2$ où $A = \left(a_{ij}\right)$ et $B = \left(b_{ij}\right)$ sont deux matrices comportant *n* lignes et *p* colonnes.
5. Ecrire une fonction qui renvoie un motif de *A* le plus proche d'une matrice *B* au sens des moindres carrés.

4. Centrale Supélec 2015 Exercice type N°1 (PSI) ♠♠

Pour un entier $n \in \mathbb{N}^*$, on considère la matrice carrée $M(n)$ formée « en serpent » par les nombres 1, 2, 3, 4, …, n^2. Par exemple :

$$M(2) = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix} \qquad M(3) = \begin{pmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 7 & 8 & 9 \end{pmatrix} \qquad M(4) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 8 & 7 & 6 & 5 \\ 9 & 10 & 11 & 12 \\ 16 & 15 & 14 & 13 \end{pmatrix}$$

1. Donner en Python ou en Scilab une fonction *f* telle que $f(n, i, j) = \left(M(n)\right)_{i,j}$.
2. Créer une fonction *M* d'argument $n \in \mathbb{N}^*$ et renvoyant $M(n)$. Tester pour $1 \leq n \leq 5$.
3. Calculer le rang de $M(n)$ pour $1 \leq n \leq 10$.
4. Conjecturer la valeur de $\mathrm{rg}\left(M(n)\right)$ pour tout $n \in \mathbb{N}^*$ et démontrer cette conjecture.

5. Définir une fonction permettant d'afficher la ligne brisée formée par les points de coordonnées $\left(k, \mathrm{tr}\left(M\left(k\right)\right)\right)$ pour $1 \leq k \leq n$.
   Tester pour $n = 100$. Essayer pour $n = 1000$.
6. Afficher les 100 premières valeurs de $\dfrac{\mathrm{tr}\left(M\left(n\right)\right)}{n^3}$. Commenter.
7. Trouver un équivalent de $\mathrm{tr}\left(M\left(n\right)\right)$ quand n tend vers $+\infty$.
8. Trouver une expression pour $\mathrm{tr}\left(M\left(n\right)\right)$ (on pourra commencer par traiter le cas où n est pair).

5. ENSAM 2015 (PT) ♠

Avec Python, on stocke les coefficients d'un polynôme $P\left(X\right) = \sum_{i=0}^{n} a_i X^i$ dans la liste $L = \left[a_n, a_{n-1}, ..., a_1, a_0\right]$.

1. Ecrire une fonction `valpol(L,x)` qui évalue P en x.
2. A l'aide de `numpy`, écrire une matrice de taille 4 à coefficients non nuls puis, en utilisant les fonctions `eye()` et `trace()` calculer $A\left(A - tr\left(A\right)I_4\right)$.
3. On donne $B_0 = M$ de taille 4 et $B_{k+1} = M\left(B_k + \dfrac{-tr\left(B_k\right)}{k+1} I_4\right)$. Construire la liste L des coefficients de $P\left(X\right) = X^n + \sum_{k=1}^{n} \dfrac{-tr\left(B_{k-1}\right)}{k} X^{n-k}$.
4. Tracer la courbe représentative de P.