EXERCICE 1

1. Écrire une fonction `diag` qui prend en argument une matrice et qui teste si celle-ci est diagonale ou non en renvoyant un booléen.
2. Écrire une fonction `triangSup` qui prend en argument une matrice et qui teste si celle-ci est ou n'est pas triangulaire supérieure en renvoyant un booléen.
3. Écrire une fonction `symetrique` qui prend en argument une matrice et qui teste si celle-ci est ou n'est pas symétrique en renvoyant un booléen.
4. En utilisant les fonctions `triangSup` et `symetrique` trouver et mettre en oeuvre une autre fonction `diag2` pour déterminer si une matrice est diagonale.

EXERCICE 2

On note $M_{a,b}$ la matrice $\begin{pmatrix} a & b & \dots & b \\ b & a & \ddots & b \\ \vdots & & \ddots & \vdots \\ b & \dots & \dots & a \end{pmatrix}$, où a et b sont des réels.

1. A l'aide de boucles imbriquées, écrire une fonction `MatAB` qui prend en argument a, b et n (la taille de la matrice), et qui crée cette matrice.
2. Exprimer $M_{a,b}$ en fonction de la matrice identité et de la matrice U qui ne contient que des 1.
3. Déterminer alors une deuxième méthode pour créer la matrice ci-dessus en utilisant des fonctions élémentaires de la bibliothèque numpy.

EXERCICE 3

Définition : Une matrice stochastique est une matrice carrée à coefficients positifs dont la somme des coefficients de chaque ligne vaut 1.

1. Écrire une fonction `stochastique` qui teste si une matrice est stochastique en renvoyant un booléen.
2. On considère la matrice colonne $U = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$. Soit M une matrice à coefficients positifs.

   Montrer que M est stochastique si et seulement si $MU = U$.
3. En déduire que le produit de deux matrices stochastiques est stochastique.

EXERCICE 4

On considère la suite récurrente linéaire d'ordre 3 définie par :

$$u_0 = 1, u_1 = 0, u_2 = -1 \text{ et } \forall n \in \mathbb{N},\ u_{n+3} = 2u_{n+2} - u_{n+1} + u_n$$

On considère, pour tout $n \in \mathbb{N}$, la matrice colonne $X_n = \begin{pmatrix} u_n \\ u_{n+1} \\ u_{n+2} \end{pmatrix}$.

1. Déterminer une matrice M tel que $\forall n \in \mathbb{N},\ X_{n+1} = MX_n$.
2. En déduire par récurrence que $\forall n \in \mathbb{N},\ X_n = M^n X_0$
3. Écrire une fonction python `suite(n)` qui renvoie la valeur de u_n.

EXERCICE 5

n étant un entier positif et $a_0, a_1, ...a_n$ des réels, on considère la fonction polynomiale P définie pour tout réel x par :

$$P(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} + a_n x^n$$

On la représente en python par la liste de ses coefficients. P sera donc représentée par la liste

$$[a_0, a_1, \ldots, a_{n-1}, a_n]$$

1. Écrire une fonction `polyMat` qui prend en argument une matrice carrée M de taille $m \geqslant 1$ et une fonction polynomiale P définie comme ci-dessus et qui renvoie la matrice $P(M)$ définie par
$$P(M) = a_0 I_m + a_1 M + \ldots + a_{n-1}M^{n-1} + a_n M^n$$
où I_m est la matrice identité d'ordre m.

On considère la matrice $A = \begin{pmatrix} 3 & 0 & 1 \\ -1 & 2 & -1 \\ -2 & 0 & 0 \end{pmatrix}$

2. Grâce à la fonction `polyMat` calculer $\mathrm{A}^2 - 3\mathrm{A}$ et en déduire que A est inversible.

Soit $B = \begin{pmatrix} 6 & 10 & 11 \\ 2 & 6 & 5 \\ -4 & -8 & -8 \end{pmatrix}$.

3. Calculer $B^3 - 4B^2 + 4B$.
4. En déduire que B n'est pas inversible.

EXERCICE 6

Soient les trois matrices carrées $A = \begin{pmatrix} 1 & -2 & -6 \\ -3 & 2 & 9 \\ 2 & 0 & -3 \end{pmatrix}$, $B = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$ et $C = \begin{pmatrix} 5 & -4 \\ 4 & -3 \end{pmatrix}$.

1. Écrire une fonction Python `puissances` qui prend en arguments une matrice carrée M ainsi qu'un entier naturel non nul n, et qui affiche les matrices M^k pour l'entier k variant de 1 à n.
2. Afficher les premières puissances des trois matrices A, B et C et conjecturer à chaque fois une éventuelle formule pour les puissances de chacune de ces trois matrices.
3. Démontrer les trois conjectures précédentes. (Pour les puissances de la matrice C, on fera le lien entre les matrices B et C et on utilisera la formule du binôme de Newton.)

EXERCICE 7

On considère le système linéaire de trois équations à trois inconnues $(S) : \left\{ \begin{array}{c} 2x + y = 7 \\ -3x + z = -8. \\ y + 2z = -3 \end{array} \right.$.

On admet que ce système est de Cramer.

a) Écrire le système (S) sous forme matricielle.

b) Écrire une fonction Python d'entête `Cramer` qui prend en arguments d'entrée une matrice carrée A ainsi qu'une matrice colonne B (de taille compatible), et qui renvoie la solution X du système $AX = B$.

Pour tester cette fonction, on admet que la solution du système (S) est le triplet $\left(\frac{3}{2}; 4; -\frac{7}{2}\right)$.