Exercice 1

Ecrire une fonction **partition(n)** qui construit un vecteur (numpy) x comportant n composantes telles que : $x_0 = 0 \leq x_1 \leq \cdots \leq x_{n-1} = 1$ où les $x_i,\ i \in [[1, n-2]]$ sont obtenus au hasard sur $[0, 1]$.
vous pouvez utiliser les fonctions **random.rand, concatenate** et la méthode **sort()** de numpy.

Exercice 2

Ecrire une fonction **def math_lapl(n) :** qui construit la matrice d'ordre n suivante :

$$\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

Exercice 3

On représente une permutation σ de S_n par un tableau A contenant tous les entiers de 1 à n dans un ordre quelconque. $A[i]$ repésente $\sigma(i+1)$. Ecrire une fonction **signature(A)** retournant la signature de la permutation représentée par le tableau A.

Exercice 4

Ecrire une fonction **def pivot-nonnul(A,i) :** et une fonction **def pivot-max(A,i) :** (sans regarder le cours) qui calcule le premier terme non nul (respectivement le plus grand) $|a_{ji}|,\ j \geq i$ de la matrice A.

Exercice 5

Ecire une fonction **def inverse(A) :** qui inverse une matrice inversible en transformant A en la matrice identité par opérations élémentaires sur les lignes en appliquant simultanément les mêmes opérations sur les lignes de la matrices identité. Utiliser les fonction **pivot-nonnul**, **def pivot-max(A,i) :** et **def echange(A,i,j)**

dans l'algorithme suivant L représentent les lignes de A et M celles de I

La première étape se passe comme pour la résolution d'un système :

pour i de 0 à n - 2 faire
Trouver j entre i et n - 1 tel que a_{ji} soit non nulle.

Echanger L_i et L_j .
Echanger M_i et M_j

pour m de i + 1 à n-1 faire
$L_m \leftarrow L_m - \dfrac{a_{mi}}{a_{ii}} L_i$ # mettre à zéro les coefficients en position $(m; i)$

$M_m \leftarrow M_m - \dfrac{a_{mi}}{a_{ii}} M_i$

A est alors transformée en une matrice triangulaire supérieure
I est alors transformée en une matrice triangulaire inférieure
Deuxième etape.

pour k de n-2 à 1 faire
Trouver l entre n - 1 et k tel que a_{lk} soit non nulle.
Echanger L_k et L_l
Echanger M_k et M_l

pour m entre k-2 et 0 faire
$L_m \leftarrow L_m - \frac{a_{mk}}{a_{kk}} L_k$
$M_m \leftarrow M_m - \frac{a_{mk}}{a_{kk}} M_k$

pour i entre 0 et n-1 faire
pour j entre 0 et n-1 faire
$M_i \leftarrow \frac{M_i}{a_{ii}}$

retourner M.

Exercice 6

Lorsqu'une matrice inversible A ne nécessite aucun échange de ligne dans la méthode de Gauss il est possible d'écrire $A = LU$ où L (respectivement U) est une matrice triangulire inférieure (respectivement triangulaire supérieure). ce résultat est obtenu par la méthode de Gauss.
Lorsque par opérations sur les lignes (sans échange) la matrice A est transformée en matrice triangulaire supérieure (c'est U) la matrice I est transformée par les mêmes opérations en L^{-1}. Ecrire une fonction **def gauss_lu(A)** qui renvoie les deux matrices L et U.

Exercice 7

Quand la matrice A est sous la forme LU, résoudre un système $Ax = b$ revient à résoudre deux systèmes simples :$Ly = b$ puis $Ux = y$. Ecrire deux fonctions **def trian_inf(L,y) :** et **def trian_sup(M,x) :** réalisant ces résolutions.