Arithmétique ou Pivot de Gauss

1 Arithmétique

1. Ecrire une première fonction, $premier(n)$, qui renvoie $True$ si l'entier n est premier, $False$ sinon.
   NB : on pensera à l'astuce de la racine carrée pour ne pas avoir trop de tests à faire.
2. **Crible d'Eratosthène**
   Ecrire alors naîvement une fonction $premiersinf(n)$ qui renvoie la liste des nombres premiers compris entre 2 et n.
3. Définir alors une fonction $eratos(n)$ qui réalise le crible d'Eratosthène à partir de la liste des entiers de 2 à n.
   Indications :
   — créer une variable p qui prend pour valeur le dernier nombre premier que l'on a entouré (on commence donc en initialisant celle-ci à 2), et une variable avec la listedes entiers de 2 à n.
   — faire une boucle *while* qui s'arrêtera quand ce nombre p est strictement supérieur à la racine carrée de n, dans laquelle :
   — on éliminera alors tous les multiples de p (sauf p) de la liste,
   — on déterminera le nombre premier qui suit p (c'est le plus petit nombre qui est encore dans la liste et qui est plus grand que p) et l'on affectera ce nombre à la variable p
4. Challenge : améliorer suffisament les fonctions précédentes pour obtenir en quelques secondes la liste des nombres premiers entre 1 et 10^6.
5. **Décomposition en produit de facteurs premiers.**
   Ecrire une fonction $decomp(n)$ qui renvoie la décomposition de l'entier n sous la forme d'une liste de listes à deux éléments correspondant à chaque nombre premier qui y apparaît assorti de la puissance à laquelle il y est :
   $$360 = 2^3 3^2 5^1 \text{ donc } decomp(360) \text{ renverra } [[2,3],[3,2],[5,1]].$$

2 Pivot de Gauss

On va stocker et manipuler dans cette séance des matrices stockées sous forme d'une liste des lignes de la matrice, chaque ligne étant simplement la liste de ses éléments.
Par exemple, la matrice :

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

sera stockée sous la forme :

```
matrice=[[1,2,3],[4,5,6]]
```

Pour accéder à l'élément de la deuxième ligne et troisième colonne, on fait :

```
matrice[1][2]
```

car la deuxième ligne a pour numéro 1 et la troisième colonne est numérotée 2.
En s'aidant par exemple de la description ci-dessous issue de wikipedia, programmer une fonction python, *pivot*(*matrice*) qui prend en argument une matrice ainsi définie, *matrice*, et qui renvoie sa forme échelonnée réduite.

```
Gauss-Jordan
  r = 0
                          (r est l'indice de ligne du dernier pivot trouvé)
  Pour j de 1 jusqu'à m            (j décrit tous les indices de colonnes)
  |   Rechercher max(|A[i,j]|, i entre r+1 et n).
  |              Noter k l'indice de ligne du maximum
  |                                          (A[k,j] est le pivot)
  |   Si A[k,j] différent de 0 alors
  |           (A[k,j] désigne la valeur de la ligne k et de la colonne j)
  |   |   r=r+1
  |                      (r désigne l'indice de la future ligne servant de pivot)
  |   |   Diviser la ligne k par A[k,j]
  |     (On normalise la ligne de pivot de façon que le pivot prenne la valeur 1)
  |   |   Si k différent de r alors
  |   |       |   échanger les lignes k et r
  |                                    (On place la ligne du pivot en position r)
  |   |   Fin Si
  |   |   Pour i de 1 jusqu'à n                (On simplifie les autres lignes)
  |   |   |   Si i différent de r alors
  |   |   |   |   Soustraire à la ligne i la ligne r multipliée par A[i,j]
                                                    (de façon à annuler A[i,j])
  |   |   |   Fin Si
  |   |   Fin Pour
  |   Fin Si
  Fin Pour
Fin Gauss-Jordan
```