Pour s'exercer...

Exercice 1
Triangle de Pascal.

On rappelle la formule de Pascal : pour tous $n, k \in \mathbb{N}$, $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$.

De plus, pour tout $n \in \mathbb{N}$, $\binom{n}{0} = 1$

Ecrire une fonction `triangle_pascal` qui prend en entrée un entier $n \in \mathbb{N}$ et renvoie la matrice A contenant le triangle de Pascal jusqu'à la ligne n. Autrement dit :

$$\text{Pour tout } (i,j) \in [\![0,n]\!]^2, \quad \texttt{A[i,j]} = \binom{i}{j}$$

Pour ce faire, on pourra :

$\boxed{1}$ Définir une matrice `A` de la bonne taille, contenant uniquement des 0.

$\boxed{2}$ Remplir correctement la 0-ième colonne de A.
La 0-ième ligne est alors correctement remplie.

$\boxed{2}$ A partir de la ligne 1, utiliser la formule de Pascal pour définir `A[i,j]` à l'aide des coefficients de la ligne précédente. On écrira deux boucles `for` imbriquées.

```
import numpy as np

def triangle_pascal(n) :

  A = np.zeros((n+1,n+1))
  A[:,0] = np.ones(n+1)   # (ou simplement : A[:,0] = 1)

  for i in range(1,n+1) : # i = 1,2, ... , n
    for j in range(1,n+1) : # j = 1, 2, ... , n :
      A[i,j] = A[i-1,j-1] + A[i-1,j]

  return A
```

Afficher ainsi `triangle_pascal(10)`.

Exercice 2
Pivot de Gauss.

1. Compléter le programme suivant pour que la fonction `echange(A,i,j)` renvoie la matrice Aà ? laquelle on a échangé les lignes i et j.

Par exemple, `echange(np.eye(3),1,2)` doit renvoyer : $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

```
import numpy as np
def echange(A,i,j) :

        L = A[i-1,:].copy()

        A[i-1, :] = A[j-1,:]

        A[j-1,:] = L

        return(A)
```

2. Compléter le programme suivant pour que la fonction `operation(A,i,j,b)` renvoie la matrice A ? laquelle on a effectué l'opération $L_i \leftarrow L_i + bL_j$.

Par exemple, `operation(np.eye(3),2,1,2)` doit renvoyer : $\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

On définira cette fonction ? la suite de la précédente.

```
def operation(A,i,j,b) :

        A[i-1, : ] = A[i-1, :] + b * A[j-1,:]
        return(A)
```

3. Définir dans la console la matrice $A = \begin{pmatrix} 3 & 1 & -1 \\ 2 & 1 & -1 \\ 1 & 1 & 2 \end{pmatrix}$.

En utilisant des opérations `A = echange(A,i,j)` et `A = operation(A,i,j,b)` successivement, appliquer l'algorithme du pivot de Gauss pour transformer A en une matrice triangulaire supérieure. On affichera A entre deux instructions pour contr ?ler le résultat de l'opération effectuée.

Matrice obtenue à la fin du pivot de Gauss : ....

Pourquoi peut-on en déduire que A est inversible ?