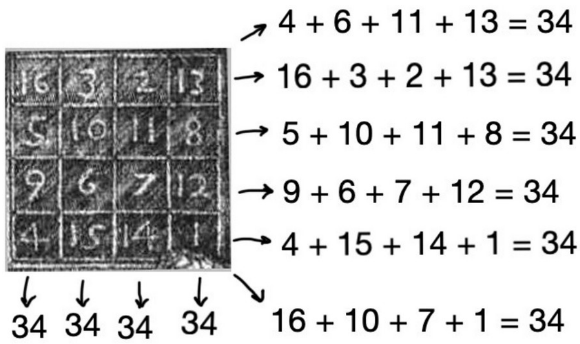
NOM, Prénom :	26/03/24 **Durée : 45 min**
DS 3 Algorithmique et Informatique	**TB1**

Si au cours du devoir, vous repérez ce qui vous semble être une erreur d'énoncé, vous le signalez sur votre copie et vous poursuivez en expliquant les raisons des initiatives que vous avez été amené à prendre.
L'usage d'une calculatrice est interdit pour ce devoir.

Un carré magique d'ordre n est composé de n^2 entiers strictement positifs, écrits sous la forme d'un tableau carré. Ces nombres sont disposés de sorte que leurs sommes sur chaque rangée, sur chaque colonne et sur chaque diagonale principale soient égales.

Ci-dessous un exemple célèbre de carré magique d'ordre 4 issu d'une gravure d'Albrecht Dürer (1514) :



Partie I Savoir si un carré est magique ou pas

On assimile un carré d'ordre n à une matrice carrée de dimension $n \times n$.

1. On donne le script suivant où M est une matrice carrée de dimension 3×3 à coefficients entiers strictement positifs :

```
def S(M):
    L=0
    for i in range(3):
        L=L+M[i][i]
    return L
```

a) De quel type est la variable M ? la variable L ?

M est de type array ou liste de listes

L est de type entier

b) Si $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$, que vaut M[1][1] ? M[2][0] ?

M [1][1] = 5

M [2][0] = 7

c) Que renvoie la fonction S ainsi définie pour $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$? Quel est le rôle de la fonction S ?

$S(M) = 1+5+9 = 15$

On additionne les coefficients diagonaux M [i][i].

Le rôle de S est donc de calculer la somme des coefficients diagonaux de M.

Remarque *ce nombre est appelé la trace de* M *et se note* $tr(M)$ *.*

d) Compléter le script suivant pour généraliser le précédent à des matrices carrées de dimension $n \times n$

.

```
def S_general(M):

    dim=M.shape[0]
    L=0
    for i in range(dim):
        L=L+M[i][i]
    return L
```

2. En vous inspirant du script de la fonction **S**, créer une fonction **somme_ligne** qui prend pour arguments une matrice carrée M de dimension 3×3 et le rang i d'une de ses lignes et qui renvoie la somme des coefficients de la ligne i de la matrice M.

   Par exemple : si $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ alors **somme_ligne(M,1)** renvoie 15

```
def somme_ligne(M,i):

    dim = M.shape[0]
    S = 0

    for j in range(dim):
        S = S + M[i][j]

    return S
```

3. On admet que l'on dispose :

   - d'une fonction **somme_colonne** qui prend pour arguments une matrice carrée M de dimension 3×3 et le rang j d'une de ses colonnes et qui renvoie la somme des coefficients de la colonne j de la matrice M.

     Par exemple : si $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ alors **somme_colonne(M,2)** renvoie **18**

   - d'une fonction **somme_diagonales** qui prend pour arguments une matrice carrée M de dimension 3×3 qui renvoie la somme des coefficients de chacune des diagonales sous la forme d'une liste.

     Par exemple : si $M = \begin{pmatrix} 1 & 0 & 1 \\ 4 & 0 & 6 \\ 5 & 1 & 9 \end{pmatrix}$ alors **somme_diagonales(M)** renvoie **[10,6]**

Compléter le script suivant :

```python
def est_magique(M):

    out = True
    S = somme_ligne(M,0)

    for i in range(1,3):
        if somme_ligne(M,i) != S :
            print(somme_ligne(M,i))
            out = False


    for i in range(3):
        if somme_colonne(M,i) != S :
            print(somme_colonne(M,i))
            out = False

    if somme_diagonales(M) != [S,S]:
       print(somme_diagonales(M))
       out = False

    return out
```

où la fonction **`est_magique`** prend pour argument une matrice carrée de dimension renvoie **`True`** si représente un carré magique, **`False`** sinon.

II   Construction d'un carré magique de dimension 3 :   la méthode de Lucas

Au XIXe siècle, Édouard Lucas a trouvé une formule générale pour construire les carrés magiques d'ordre 3.

Il a constaté qu’avec a, b et c des entiers tels que : $0 < a < b < c - a$ et $b \neq 2a$, le carré suivant est un carré magique :

$c+a$	$c-a-b$	$c+b$
$c-a+b$	c	$c+a-b$
$c-b$	$c+a+b$	$c-a$

Définir une fonction **`carre_magique`** qui prend comme arguments trois entiers a, b et c et qui renvoie un carré magique (donc une matrice M) si a, b et c vérifient $0<a<b<c-a$ et $b \neq 2a$ et renvoie un message d'erreur sinon.
On pourra utiliser la fonction zeros(n,p) qui renvoie un array de dimension * dont les coefficients sont tous nuls.

Vous pourrez éventuellement proposer un algorithme en langage naturel.

Par exemple :

```python
def magique(a,b,c):

    if a<b<c-a and b!=2*a:

        M=np.zeros((3,3))
        M[0][0]=c+a
        M[0][1]=c-a-b
        M[0][2]=c+b
        M[1][0]=c-a+b
        M[1][1]=c
        M[1][2]=c+a-b
        M[2][0]=c-b
        M[2][1]=c+a+b
        M[2][2]=c-a

        return M

    return False
```

BONUS

Écrire le script de la fonction **`somme_diagonales`** définie au I.3

```python
def Somme_diag(M):
    L=[0,0]
    for i in range(dim):
        L[0]=L[0]+M[i][i]   #  calcule la somme des coefficients de la 1ère diagonale
                            # relativement facile à voir au vu des indices de M[i][i]

        L[1]=L[1]+M[i][2-i] #  calcule la somme des coefficients de la 2ème diagonale
                            # bien mettre en évidence les coefficients successifs en
                            # faisant varier i de 0 à 2
    return L
```