

"""

Author: Pascal Padilla

Source: correction de l'exercice 2 du sujet 20 des épreuves pratiques NSI 2022

Remarque:

la classe Carre a 2 propriétés et 3 méthodes:

; self.ordre: ordre du carré (et donc nb de ligne et colonne)

; self.valeurs: tableau contenant les valeurs (tableau de tableau)

il est difficile de faire les tests car il faut créer les

carrés c2, c3 et c4 avec la P00

"""

class Carre:

def \_\_init\_\_(self, tableau = []):

self.ordre = len(tableau)

self.valeurs = tableau

def affiche(self):

'''Affiche un carré'''

# affiche ligne par ligne le tableau de valeurs

for i in range(self.ordre):

print(self.valeurs[i])

def somme\_ligne(self, i):

'''Calcule la somme des valeurs de la ligne i'''

# utilise la fonction `sum` qui calcule la somme d'un tableau

return sum(self.valeurs[i])

def somme\_col(self, j):

'''calcule la somme des valeurs de la colonne j'''

# génère un tableau par compréhension pour créer la colonne j

# puis utilise la fonction sum pour calculer la somme des valeurs

return sum([self.valeurs[i][j] for i in range(self.ordre)])

def est\_magique(carre):

n = carre.ordre

s = carre.somme\_ligne(0)

#test de la somme de chaque ligne

# ; de la 2ème ligne à la (n - 1)ième ligne

for i in range(1, n):

if carre.somme\_ligne(i) != s:

return False

#test de la somme de chaque colonne

# ; utilise la méthode .somme\_col(j) pour calculer la somme de la colonne

for j in range(n):

if carre.somme\_col(j) != s:

return False

#test de la somme de chaque diagonale

# ; création d'un tableau en compréhension : tab[k][k] for k in ...

# pour avec le tableau de la diagonale

# [tab[0][0], tab[1][1], tab[2][2], ...]

if sum([carre.valeurs[k][k] for k in range(n)]) != s:

return False

if sum([carre.valeurs[k][n-1-k] for k in range(n)]) != s:

return False

# si aucun renvoie n'a eu lieu, c'est le carré est magique

# il faut renvoyer la valeur de la somme

return s

# tests

c2 = Carre([[1, 1], [1, 1]])

print(est\_magique(c2))

c3 = Carre([[2, 9, 4], [7, 5, 3], [6, 1, 8]])

print(est\_magique(c3))

```
c4 = Carre([[4, 5, 16, 9], [14, 7, 2, 11], [3, 10, 15, 6], [13, 12, 8, 1]])  
print(est_magique(c4))
```