

```

# ===== #
# MP #
# Concours blanc - 10 janvier 2018 #
# EXERCICE I #
# CARRÉS MAGIQUES D'ORDRE IMPAIR #
# ===== #

## Importations
import numpy as np

## Question 1

# Un carré magique d'ordre n comporte les  $n^2$  premiers entiers naturels non
nuls
# (de 1 à  $n^2$  donc). Leur somme vaut  $n^2(n^2+1)/2$ . Comme on a n lignes et que
la
# somme des coefficients de chaque ligne vaut S, il vient  $nS=n^2(n^2+1)/2$ ,
soit,
# finalement,  $S=n(n^2+1)/2$ .

## Question 2
# ATTENTION ! Une seule instruction était demandée !

# L1 = list(range(1,n**2+1))

# ou :

# L2 = [k for k in range(1,n**2+1)]

## Question 3 - Fonction de validation
def MagicSquareValid(M):
    (n,p) = M.shape
    # Le tableau est-il carré ?
    if n != p:
        print("Le \"carré\"... n'en est pas un !")
        return(False)
    else:
        S = n*(n**2 + 1)//2
        # Liste des entiers de 1 à  $n^2$ 
        L = list(range(1,n**2 + 1))
        # Tous les entiers de 1 à  $n^2$  sont-ils dans M ?
        # On transforme le tableau M en une liste (M2) de  $n^2$  éléments
        M2 = list(M.reshape(1,n**2)[0])
        # On trie cette liste
        M2.sort()
        # Cas où M2 n'est pas identique à L
        if M2 != L:
            print("M ne contient pas tous les entiers de 1 à  $n^2$ ")
            return(False)
        # La somme des coefficients de la ligne i vaut-elle bien S ?
        for i in range(n):
            if sum(M[i,:]) != S: # On peut aussi écrire M[i]
                print("La somme des coefficients de la ligne "+str(i)+" ne
vaut pas "+str(S))
                return(False)
        # La somme des coefficients de la colonne j vaut-elle bien S ?
        for j in range(n):
            if sum(M[:,j]) != S:
                print("La somme des coefficients de la colonne "+str(j)+" ne
vaut pas "+str(S))

```

```

        return(False)
    # La somme des coefficients de la diagonale principale vaut-elle
    bien S ?
    SD = 0
    for k in range(n):
        SD += M[k,k]
    if SD != S:
        print("La somme des coefficients de la diagonale principale ne
    vaut pas",S)
        return(False)
    # La somme des coefficients de la seconde diagonale principale vaut-
    elle bien S ?
    SD2 = 0
    for k in range(n):
        SD2 += M[k,n - k - 1]
    if SD2 != S:
        print("La somme des coefficients de la diagonale principale ne
    vaut pas",S)
        return(False)
    # Tous les tests sont concluants
    return(True)

```

### ## Questions 4 à 7 - Fonction CM

```

def CM(n):
    # Matrice de zéros d'ordre 2n-1 (Question 4)
    M = np.zeros((2 * n - 1, 2 * n - 1),dtype=int)

    # Remplissage des diagonales (Question 5)
    # Il y a n diagonales --> compteur d
    for d in range(n):
        # Il y a n éléments dans chaque diagonale --> compteur e
        for e in range(n):
            M[d + e,n - d - 1 + e] = 1 + d * n + e

    # Une autre possibilité :
    #c = 1
    #for d in range(n):
    #    for e in range(n):
    #        M[d + e,n - d - 1 + e] = c
    #        c += 1

    # Déplacement des blocs (Question 6)
    k = n // 2
    # --> bloc du haut (va en bas dans le carré magique)<-- #
    M[n:3 * k + 1,k + 1:3 * k] += M[:k,k + 1:3 * k]
    # --> bloc du bas (va en haut dans le carré magique) <-- #
    M[k:2 * k,k + 1:3 * k] += M[3 * k + 1:,k + 1:3 * k]
    # --> bloc de gauche (va à droite dans le carré magique) <-- #
    M[k + 1:3 * k,n:3 * k + 1] += M[k + 1:3 * k,:k]
    # --> bloc de droite (va à gauche dans le carré magique) <-- #
    M[k + 1:3 * k,k:2 * k] += M[k + 1:3 * k,3 * k + 1:]

    # Affichage d'une info complémentaire
    print("La somme de chaque ligne, chaque colonne et chaque diagonale
    principale vaut :",n * (n**2 + 1) // 2)

    # renvoi du carré comme sous-tableau (Question 7)
    return(M[k:3 * k + 1,k:3 * k + 1])

```

## Pour tester...

```
n = 4
while not(n%2):
    n = int(input("Veuillez saisir l'ordre de votre carré : "))
print(CM(n))
```