

Exercice corrigé : Carré magique

Un carré magique d'ordre n est un tableau carré $n \times n$ dans lequel on écrit une et une seule fois les nombres entiers de 1 à n^2 , de sorte que la somme des n nombres de chaque ligne, colonne ou diagonale principale soit constante. P.ex. le carré magique d'ordre 5, où toutes les sommes sont égales à 65 :

11	18	25	2	9
10	12	19	21	3
4	6	13	20	22
23	5	7	14	16
17	24	1	8	15

Pour les carrés magiques d'ordre impair, on dispose de l'algorithme suivant – (i,j) désignant la case de la ligne i , colonne j du carré ; on se place en outre dans une indexation « naturelle » commençant à 1 :

1. la case $(n, (n+1)/2)$ contient 1 ;
2. si la case (i,j) contient la valeur k , alors on place la valeur $k+1$ dans la case $(i+1, j+1)$ si cette case est vide, ou dans la case $(i-1, j)$ sinon. On respecte la règle selon laquelle un indice supérieur à n est ramené à 1.

Programmer cet algorithme pour pouvoir construire un carré magique d'ordre impair quelconque.

```
# -*- coding: utf-8 -*-
```

```
Created on Tue Nov 22 08:40:40 2022
```

```
@author: chikhaoui ahmed
```

```
n = int(input(" Donner le coté du carré ")) # la valeur saisie est de type str, il faut la convertir au #type approprié
```

```
assert n % 2 == 1, "L'ordre { } n'est pas impair".format(n) # l'assertion facilite le débogage
```

```
array = [[0 for j in range(n)] for i in range(n)] # Initialisation du tableau: liste de n listes de n #zéros.
```

```
i, j = n, int((n + 1) / 2) # Initialisation de l'algorithme
```

```
array[i - 1][j - 1] = 1
```

```
for k in range(2, n ** 2 + 1):
```

```
    ik = (i + 1)%n
```

```
    jk = (j + 1)%n
```

```
    if array[ik - 1][jk - 1] == 0:
```

```
        i, j = ik, jk
```

else:

```
i = (i - 1) % n
```

```
array[i - 1][j - 1] = k
```

```
print("Carré magique d'ordre {} :".format(n))
```

```
for row in array:
```

```
    print(' '.join('{:2d} '.format(k) for k in row), '=', sum(row))
```

```
    print(' '.join('==' for k in row))
```