

TD 2 - Fonctions, modules, exceptions

Ex1- Suite de Fibonacci

Fonctions simples

1. Ecrire une fonction qui affiche la suite de Fibonacci jusqu'à une limite passée en paramètre.
2. Ecrire une fonction qui retourne la suite de Fibonacci jusqu'à une limite passée en paramètre.
3. Ecrire le programme principal qui appelle ces 2 fonctions après avoir demandé à l'utilisateur la valeur limite

Ex2- Couicable

Fonctions simples, décorateur, retour d'une ou plusieurs valeurs

1. Écrire la fonction `somme_chiffre()` qui retourne la somme des chiffres composant le nombre passé en paramètre.
2. Écrire la fonction `nombre_chiffres()` qui retourne le nombre des chiffres composant le nombre passé en paramètre.
3. Écrire la fonction `separe_nombre()` qui sépare le nombre passé en paramètre en 2 : partie droite et gauche.
4. Écrire un décorateur `test_nombre_chiffres_pair()` permettant de tester que l'argument de la fonction (dans cet exercice `separe_nombre()`) passée est bien un entier composé d'un nombre de chiffres pair. Si ce n'est pas le cas, la fonction renverra (0,0)
5. Écrire le programme principal qui demande un nombre à l'utilisateur et affiche si celui-ci est couicable ou non. Un nombre est dit couicable s'il est composé d'un nombre de chiffres pair et si la sommes des chiffres de sa partie droite est égale à la somme des chiffres de sa partie gauche.
6. Écrire la fonction `somme_chiffre()` mais cette fois-ci de façon récursive.

Exo3 - Carré magique

Fonctions simples, définition et utilisation d'un module, *args, listes de listes

Nous voudrions définir un module `carré_magique` qui contient les fonctions relatives aux carrés magiques. Les fonctions demandées dans les questions 1 à 6 seront définies dans ce module. La question 7 sera définie dans un fichier contenant aussi la fonction `main()` qui utilisera ce module.

Définition d'un carré magique

On considère un entier n strictement positif. Un carré magique d'ordre n est une matrice carrée d'ordre n (n lignes et n colonnes), qui contient des nombres entiers strictement positifs. Ces nombres sont disposés de sorte que les sommes sur chaque ligne, les sommes sur chaque colonne et les sommes sur chaque diagonale principale soient égales. La valeur de ces sommes est appelée : constante magique. Exemple : Carré magique d'ordre 3, sa constante magique vaut 45

21	7	17	→45
11	15	19	→45
13	23	9	→45
↙45	↓45	↓45	↘45

1. Écrire la fonction `somme_ligne(mat,i)` , qui reçoit en paramètres une matrice carrée `mat` contenant des nombres, et un entier `i` qui représente l'indice d'une ligne dans `mat`. La fonction retourne la somme des nombres de la ligne d'indice `i` dans `mat`.
2. Écrire la fonction `somme_colonne(mat,j)` , qui reçoit en paramètres une matrice carrée `mat` contenant des nombres, et un entier `j` qui représente l'indice d'une colonne dans `mat`. La fonction retourne la somme des nombres de la colonne `j` dans `mat`.
3. Écrire la fonction `somme_diag1(mat)` , qui reçoit en paramètre une matrice carrée `mat` contenant des nombres, et qui retourne la somme des éléments de la première diagonale principale dans `mat`.
4. Écrire la fonction `somme_diag2(mat)` , qui reçoit en paramètre une matrice carrée `mat` contenant des nombres, et qui retourne la somme des éléments de la deuxième diagonale principale dans `mat`.
5. Écrire la fonction `magique(mat_c)` , qui reçoit en paramètre une matrice carrée `mat_c` contenant des entiers strictement positifs, et qui retourne `True`, si la matrice `mat_c` est un carré magique et `False`, sinon.

Un carré magique normal d'ordre n est un carré magique d'ordre n , constitué de tous les nombres entiers positifs compris entre 1 et n à la puissance 2 .

Exemple : Carré magique normal d'ordre 4, composé des nombres entiers : 1, 2, 3, ..., 15, 16.

4	14	15	1
9	7	6	12
5	11	10	8
16	2	3	13

6. Écrire la fonction `carré_magique_normal(mat_c)` , qui reçoit en paramètre une matrice carrée `mat_c` qui représente un carré magique. La fonction retourne `True` si le carré magique `mat_c` est normal, sinon, elle retourne `False` .

Exemples

La fonction `carré_magique_normal([[8, 1, 6] , [3, 5, 7] , [4, 9, 2]])` retourne `True` .

La fonction `carre_magique_normal([[21, 7, 17] , [11, 15, 19] , [13, 23, 9]])` retourne `False`

7. Dans une autre fichier, écrire une fonction `affiche_test_cm(*args)` qui reçoit comme paramètres une ou plusieurs matrices, les teste et affiche si elles sont magiques et le cas échéant si elles sont "normales". Écrire le programme principal qui permet de tester les exemples ci-dessous.

```
C1= [[8, 1, 6], [3, 5, 7], [4, 9, 2]]
C2 = [[21,7,17],[11,15,19],[13,23,9]]
C3= [[30,39,48,1,10,19,28], [38,47,7,9,18,27,29],
      [46,6,8,17,26,35,37], [5,14,16,25,34,36,45],
      [13,15,24,33,42,44,4], [21,23,32,41,43,3,12],
      [22,31,40,49,2 ,11,20]]
```