

Exercice : Fonction en Python

Exercice 1

On donne le script python suivant.

```
def ligne(n,i):  
    motif="0"*i+"1"*(n-2*i)+"0"*i+"\n"  
    return motif  
  
def pointe(n):  
    dessin=""  
    for i in range(n//2+1):  
        dessin+=ligne(n,i)  
    return dessin  
  
figure=pointe(7)  
print(figure)
```

- 1) Que renvoie l'appel de la fonction **ligne(10,3)** ?
- 2) On exécute le script. Quel est l'affichage obtenu ?

Exercice 2

- 1) Créer une fonction "somme_carre" qui prend en paramètre deux nombres a et b et renvoie la somme des carrés des deux nombres.
- 2) Créer une fonction "double_produit" qui prend en paramètre deux nombres a et b et renvoie le double du produit de ces deux nombres.
- 3) On veut créer une fonction id_rem qui prend en paramètre deux nombres a et b et renvoie le carré de la somme de ces deux nombres.
 - a) Écrire cette fonction.
 - b) Écrire cette fonction en utilisant les fonctions somme_carre et double_produit.

Exercice 3

- 1) Créer une fonction "somme_arithm" qui prend en paramètre un nombre entier N et retourne la somme de tous les nombres entiers inférieurs ou égaux à N.
- 2) Créer une fonction "somme_carre" qui prend en paramètre un nombre entier N et retourne la somme des carrés des nombres entiers inférieurs ou égaux à N.
- 3) Créer une fonction "somme_inverse" qui prend en paramètre un nombre entier N et retourne la somme des inverses des nombres entiers inférieurs ou égaux à N.
- 4) Créer une fonction "somme_inverse_carre" qui prend en paramètre un nombre entier N et retourne la somme des carrés des inverses des nombres entiers inférieurs ou égaux à N.

Exercice 4

- 1) Écrire une fonction "zen" qui renvoie la ligne de 10 caractères : ZNZNZNZNZN
- 2) Modifier cette fonction pour prendre en paramètre le nombre n de caractères.
 - si n est pair, la ligne commence par Z et se finit par N ;
 - si n est impair, la ligne comment et finit par la lettre Z.
- 3) Modifier cette fonction pour prendre aussi en paramètre chaque caractère, N et Z étant les valeurs par défaut de la fonction.
- 4) Écrire un script avec la fonction zen pour reproduire la figure ci-dessous.

```
TCTCTCTCTC
TCTCTCTCT
TCTCTCTC
TCTCTCT
TCTCTC
TCTCT
TCTC
TCT
TC
T
```

- 5) Écrire une fonction "pluszen" qui utilise la fonction zen pour renvoyer p lignes et n colonnes de 2 caractères décalés à chaque ligne. Par exemple, le code ci-dessous :

```
fig=pluszen(5,4,"N","Z")
print(fig)
```

renvoie l'affichage :

```
NZNZN
ZNZNZ
NZNZN
ZNZNZ
```

Exercice 5

- 1) Créer une fonction "est_impair" qui prend en paramètre un nombre entier et renvoie vrai si le nombre est impair et faux s'il ne l'est pas.
- 2) Créer une fonction "est_multiple" qui prend en paramètre 2 nombres entiers et qui renvoie vrai si le premier nombre est un multiple du second et faux sinon.
- 3) Un nombre entier est dit **premier** s'il est divisible seulement par 1 et lui-même. Écrire une fonction "est_premier" qui prend en paramètre un nombre entier et renvoie vrai s'il est premier et faux sinon.

Exercice 6

Un palindrome est un mot qui peut se lire dans les deux sens, comme par exemple LAVAL et NON.

- 1) Créer une fonction renverse qui prend en paramètre une chaine de caractères et renvoie cette même chaine écrite dans l'autre sens.
- 2) Écrire une fonction palindrome qui prend en paramètre une chaine de caractères et renvoie vrai si cette chaine est un palindrome.
- 3) Faire de même avec les nombres entiers en écriture décimale.