

Exercices récursivité :

1. Qu'affiche le programme suivant ?

```
def dk(L1,L2=[]):
    if L1 == []:
        return L2
    else:
        s = L1.pop(0)
        if s not in L2:
            L2.append(s)
        return dk(L1,L2)
print ( dk([34,2,3,11,11,2,34,7,1,7,7,11,3,11]) )
```

2. Rendre récursive la fonction **Somme** suivante :

```
def somme(L):
    s=0
    for val in L :
        s += val
    return s
```

3. Soit la suite u définie par :

$$u_0 = 0$$

$$\forall k \geq 1 \quad u_{k+1} = 2u_k + k$$

Ecrire une fonction **u** récursive permettant de calculer le nième terme de la suite.

```
def u(n):
```

4. Ecrire une fonction récursive **sommePremiersCarres** qui calcule la somme des n premiers carrés (avec $n > 0$).

Exemple : pour $n=3$, la fonction calculera $1^2 + 2^2 + 3^2$.

```
def sommePremiersCarres(n) :
```

5. Ecrire une fonction récursive **estTrie** permettant de vérifier qu'une liste L passée en paramètre est triée ou non.

```
def estTrie(liste):
```

6. Un mot est un palindrome si on peut le lire dans les deux sens de gauche à droite et de droite à gauche. KAYAK est par exemple un palindrome.

Ecrire une fonction **estPalindromeRec** récursive permettant de vérifier si un mot est palindrome.

```
def estPalindromeRec(ch):
```

Ecrire une fonction **estPalindromelter** récursive permettant de vérifier si un mot est palindrome.

```
def estPalindromelter(ch):
```

7. On rappelle que le PGCD (plus grand diviseur commun de deux nombres) vérifie la propriété suivante : si la division euclidienne de a par b s'écrit $a = b \times q + r$, alors $\text{pgcd}(a,b) = \text{pgcd}(b,r)$. Cette propriété est à la base de l'algorithme d'Euclide.

Exemple :

$$\text{pgcd}(24,18) = \text{pgcd}(18,6) = \text{pgcd}(6,0) \text{ donc } \text{pgcd}(24,18) = 6$$

Ecrire une fonction récursive **pgcd(a,b)** :

```
def pgcd(a, b):
```

8. Écrire un algorithme récursif **recherche(liste,val)** qui recherche la présence de la valeur val dans une liste **triée** (par ordre croissant) liste. Cette fonction doit renvoyer un booléen.