

Fonctions récursives : Correction

→ Exemple. : **Ecrire** un sous-programme récursif qui calcule la somme des n premiers carrés. Par exemple, si n vaut 3, ce sous-programme calculera $1^2+2^2+3^2$

Paramètre(s) de récurrence : **Valeur de n**

Cas de base : **$n=1$**

Variation de paramètre : **$n-1$**

Programme :

```
def somme(n):  
    if n==1:  
        return 1  
    else :  
        return n**2+somme(n-1)
```

→ Exercice 1. : **Ecrire** un sous-programme récursif qui calcule la somme des éléments positifs d'un tableau.

Paramètre(s) de récurrence : **tableau**

Cas de base : **$\text{len}(\text{tableau})==1$**

Variation de paramètre : **$\text{tableau}[1:]$**

Programme :

```
def calculElemPositifs(tab) :  
    if len(tab)==1 :  
        if tab[0]>0 :  
            return tab[0]  
        else :  
            return 0  
    else :  
        if tab[0]>0 :  
            return tab[0]+calculElemPositifs(tab[1:])  
        else :  
            return calculElemPositifs(tab[1:])
```

→ Exercice 2.: **Ecrire** un sous-programme récursif qui inverse les caractères d'une chaîne de caractères.

Paramètre(s) de récurrence : **Chaîne de caractères (chaîne)**

Cas de base : **len(chaîne)==1**

Variation de paramètre : **chaîne[1:]**

Programme :

```
def inverse(chaîne):  
    if len(chaîne)==1 :  
        return chaîne  
    else :  
        return inverse(chaîne[1:])+chaîne[0]
```

→ Exercice 3.: Ecrire un sous-programme récursif qui calcule la valeur numérique d'une chaîne de caractères composée de chiffres.

Paramètre(s) de récurrence : **Chaîne de caractères (chaîne)**

Cas de base : **len(chaîne)==1**

Variation de paramètre : **chaîne[1:]**

Programme :

```
def sommeElem(chaîne):  
    if len(chaîne)==1 :  
        return int(chaîne)  
    else :  
        return int(chaîne[0])+sommeElem(chaîne[1:])
```

