

La récursivité (programmation Python)  
TD

---

**Exercice 1**

**Q.1** Écrire une fonction Python `to_string` pour convertir un entier en base 10 dans n'importe quelle base en utilisant la **récursivité**.

**Exemple:**

`to_string(2835, 16) -> B13`

**Q.2** Écrire une fonction Python récursive `sumDigits` permettant d'obtenir la somme d'un entier non négatif.

**Exemple:**

`sumDigits(345) -> 12`

`sumDigits(45) -> 9`

**Q.3** Écrire une fonction Python récursive `sum_series` permettant de calculer la somme des entiers positifs de  $n + (n - 2) + (n - 4) \dots$  (jusqu'à  $n - x \leq 0$ ).

**Exemple:**

`sum_series(6) -> 12`

`sum_series(10) -> 30`

**Exercice 2**

**Q.1** Écrire une fonction Python récursive `harmonic_sum` pour calculer la somme des séries harmoniques jusqu'à  $n$  termes..

**Note:** Note : La somme harmonique est la somme des réciproques des entiers positifs.

**Exemple :**

série harmonique  $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$

`harmonic_sum(7) -> 2.5928571428571425`

`harmonic_sum(4) -> 2.0833333333333333`

**Q.2** Écrire une fonction Python `geometric_sum` pour calculer la somme géométrique jusqu'à  $n$  termes.

**Remarque :** en mathématiques, une série géométrique est une série dont le rapport entre les termes successifs est constant.

**Exemple :**

`geometric_sum(7) -> 1.9921875`

`geometric_sum(4) -> 1.9375`

**Exercice 3**

**Q.1** Écrire un programme Python `PGCD_Rec` pour trouver le plus grand diviseur commun (PGCD) de deux nombres entiers en utilisant la récursivité.

**Exemple :**

`PGCD_Rec(12,14) -> 2`

**Q.2** Écrire un programme Python `recursive_list_sum` pour additionner des listes de récursion en utilisant la récursivité.

**Exemple :**

`recursive_list_sum([1, 2, [3,4], [5,6]]) -> 21`

**Exercice 4**

Considérons la fonction définie ci-dessous qui prend en paramètre un entier naturel  $n$ .

```
def f(n):  
    nstr=str(n)  
    s=0  
    for c in nstr:  
        s=s+int(c)  
    return s
```

**Q.1** Que renvoie l'appel  $f(52431)$  ?

**Q.2** Ecrire une fonction  $g$ , version récursive de la fonction  $f$ .

**Q.3** Ecrire une fonction récursive  $h$  qui prend en paramètre une chaîne de caractères  $c$  et renvoie la chaîne obtenue en inversant l'ordre des caractères. Par exemple  $f('abc')$  renvoie la chaîne 'cba'.

**Q.4** En déduire une fonction **renverse** qui prend en paramètre un entier naturel  $n$  et renvoie l'entier naturel obtenu en inversant l'ordre des chiffres dans l'écriture de  $n$ .

Par exemple  $renverse(5042)$  renvoie 2405; attention  $renverse(370)$  renvoie 73.

La fonction **renverse** utilise la fonction précédente  $h$