

## Algo-Langage PEC

### Feuille de TP - Séance 2 : Récursivité

1. Écrire pour chacune des fonctions suivantes une version itérative et une version récursive. Comparez le nombre de calculs effectués dans chacun des cas. Dans le cas de la version récursive, calculez également le nombre d'appels effectués.

a) fonction *factorielle*

b) fonction *somme\_chiffre* qui retourne la somme des chiffres d'un nombre dans son écriture en base 10 de manière récursive et de manière itérative.

(exemple :  $\text{somme\_chiffre}(124) = 7$ )

c) fonction *fibonacci* qui calcule le nième terme de la suite de Fibonacci définie récursivement par :

$$\text{fibonacci}(n) = \text{fibonacci}(n-1) + \text{fibonacci}(n-2) \text{ et } \text{fibonacci}(0) = \text{fibonacci}(1) = 1$$

2. Écrire une fonction qui calcule les termes de la fonction d'Ackermann et qui affiche le nombre d'appels effectués.

● Ack(m, n) vaut :

- Si  $m=0$  :  $n+1$  (*cas d'arrêt*)
- Sinon: si  $n=0$  : Ack(m-1,1)
- Sinon Ack(m-1, Ack(m,n-1))

3. Écrire des versions récursives des tris par sélection et par insertion. Comparer, en termes de rapidité et de nombre d'opérations de lecture et d'écriture, aux versions itératives de la séance précédente.
4. Écrire et comparer les versions itératives et récursives de la recherche d'un élément dans un tableau.
5. Idem pour l'inversion d'un tableau (le dernier élément devient le premier, le deuxième l'avant-dernier, etc...)