Fonctions - Récursion

Exercice 1

1. Écrire une fonction récursive qui prend comme argument une valeur entière n et qui calcule la somme des n premiers entiers. Appeler cette fonction pour qu'elle calcule la somme des entiers allant de 0 à 100.
2. Transformer cette fonction afin qu'elle admette 2 arguments `debut` et `fin` et qu'elle calcule la somme des entiers variant de la valeur entière `debut` à la valeur entière `fin`. Appeler cette fonction pour calculer la somme des entiers variant de 50 à 100.

Exercice 2

La suite de Fibonacci est définie par :

$$\text{Fib}(0) = 0 \tag{1}$$
$$\text{Fib}(1) = 1 \tag{2}$$
$$\forall\, n \geq 2,\ \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \tag{3}$$

1. Programmer une fonction itérative calculant les n premiers nombres de Fibonacci.
2. Programmer une fonction récursive calculant les n premiers nombres de Fibonacci.
3. A l'aide de la bibliothèque `time` et de sa fonction `time()`, comparer les temps de calcul de ces deux fonctions pour `n=35` puis `n=70`. Que constatez-vous ?
4. Écrire et programmer une nouvelle fonction récursive ne faisant que n appels récursifs seulement. Mesurer le temps de calcul pour `n=35` puis `n=70`. Que constatez-vous ?

Exercice 3

Écrire une fonction récursive qui prend en argument une liste d'entiers `t` et qui calcule et affiche un entier de la façon suivante : à partir de `t`, on calcule la liste des différences de 2 éléments consécutifs de `t` (à savoir `t[i+1]-t[i]`), et on recommence sur cette liste des différences jusqu'à obtenir une liste d'un seul élément. Par exemple, partant de la liste [3,5,10], la fonction doit afficher 3 (en passant par le calcul de la liste intermédiaire [2,5]).

Exercice 4

1. Écrire une fonction prenant un entier a et un entier positif n et qui retourne a^n de manière itérative.
2. Écrire une fonction récursive calculant cette même puissance.

3. Sachant que $a^0 = 1$ et

$$a^n = \begin{cases} (a^{(n/2)})^2 & \text{si } n \text{ est pair} \\ a \times (a^{(n-1)/2)})^2 & \text{si } n \text{ est impair} \end{cases}$$

écrire une deuxième fonction récursive plus maline calculant cette même puissance.

4. Afficher les puissances de 2 de 2^0 à 2^{30} en appelant les deux fonctions récursives ci-dessus. On affichera également pour chaque puissance le nombre d'appels à chaque fonction, comptabilisés à l'aide d'une variable globale.