Récursivité - Récursivité terminale

Exercice 1 : Ecriture de fonctions récursives

- Ecrire une fonction `somme` qui, appelée avec deux entiers comme arguments, renvoie la somme de tous les entiers compris au sens large entre le premier et le second : `(somme 1 5)` vaut 15, `(somme 5 1)` vaut 0.
- De même, écrire une fonction `somme-carres` qui calcule la somme des carrés des entiers situés entre ses deux arguments.
- Ecrire la fonction qui donne la valeur de la suite de Syracuse pour la valeur n sachant que :
  syracuse(1) = 0
  syracuse(n) = 1 + syracuse(n/2) si n est pair
  syracuse(n) = 1 + syracuse(1 + 3 * n) si n est impair
  (*Pour tester : syracuse(7) = 16*)
- Définir une fonction puissance suivant le principe : $x^0 = 1$, et $x^n = x.x^{n-1}$ si $n > 0$
- Réécrire la fonction puissance selon l'idée : $x^n = (x^{n/2})^2$ si n est pair, $x^n = x.x^{n-1}$ si n est impair.
- Pensez à tester vos fonctions...

Exercice 2 : Récursivité terminale

- Ecrire une fonction `plus` récursive , qui appelée avec deux entiers a et b comme arguments, effectue la somme de a et b en utilisant l'idée que pour ajouter b, on additionne "b fois" la valeur 1.
- Ecrire une fonction `produit` récursive, qui appelée avec deux entiers a et b comme arguments, effectue le produit de a par b en utilisant l'idée que $a * b$ revient à faire $a + a + ... + a + a$ (b fois).
- Rendre les deux fonctions `plus` et `produit` récursives terminales.
- Pensez à tester vos fonctions, notamment avec des valeurs "importantes"...

Exercice 3 : Fibonacci

Suite de Fibonacci : $\texttt{fibo}(n) = \texttt{fibo}(n-1) + \texttt{fibo}(n-2)$ avec $\texttt{fibo}(0) = 1$ et $\texttt{fibo}(1) = 1$.

- **Méthode 1 :** Utiliser une double récurrence. Quel est l'inconvénient majeur de cette méthode ?
- **Méthode 2 :** Utiliser une fonction annexe pour réaliser le calcul avec une simple récurrence :
  Aide : cette fonction annexe effectue le calcul d'une suite de Fibonacci généralisée ; elle prend 3 arguments : n, a et b et calcule $\texttt{fibogen}_{ab}(n) = \texttt{fibogen}_{ab}(n-1) + \texttt{fibogen}_{ab}(n-2)$ *avec* $\texttt{fibogen}_{ab}(0) = a$ *et* $\texttt{fibogen}_{ab}(1) = b$. *Pour effectuer la simple récurrence, noter que*
  (`fibogen` *0 a b) = a,*
  (`fibogen` *1 a b) = b,*
  (`fibogen` *2 a b) = a + b,*
  (`fibogen` *3 a b) = (a +b) + b = (*`fibogen` *2 b (+ a b))*
- Tester les deux versions avec la valeur 32. Que constatez-vous ? Pour confirmer cette impression, on peut utiliser la fonction `time` comme ceci :
  ```
  (time (fibo 32))
  (time (fibogen 32 1 1))
  (time (fibogen 1000 1 1))
  ```
  Que pouvez-vous dire ?

Exercice 4 : Racines - formule de Newton

- `racine-carre`
  - $a_0 = 1$
  - $a_n = \frac{1}{2}(a_{n-1} + \frac{x}{a_{n-1}})$

  Pour écrire `racine-carre`, on écrira trois fonctions :
  - la fonction `test-arret?` testant la convergence du calcul par la formule $|a_n^2 - x| < \epsilon$
  - la fonction `suivant` calculant le terme suivant de la série

  - La fonction `racine-iter` réalisant la boucle itérative
  - la fonction `racine` correspondant à l'appel initial.

- `racine-cubique`
  - $a_0 = 1$
  - $a_n = \frac{1}{3}(2a_{n-1} + \frac{x}{a_{n-1}^2})$

  On utilisera la même modularisation que pour la fonction `racine-carre`.