
Expression Logique et Fonctionnelle ... Évidemment

TD n° 3 :

Exercice 1 *Programmer des fonctions avec itere*

Il s'agit dans cet exercice de programmer des fonctions récursives classiques sans utiliser la forme **let rec**, mais en utilisant la fonction **itere**.

Question 1 La fonction **fact** : `int -> int` qui calcule la factorielle d'un entier.

Question 2 La fonction **sommel** : `int -> int -> int` qui calcule la somme des entiers compris entre les deux entiers passés en argument.

Question 3 La fonction **liste_entiers** : `int -> int list` qui construit la liste des entiers de 1 à n dans l'ordre croissant, lorsqu'elle est appliquée à un entier n .

Exercice 2 *Programmer des fonctions avec tantque*

Il s'agit dans cet exercice de programmer des fonctions récursives classiques sans utiliser la forme **let rec**, mais en utilisant la fonction **tantque**.

Question 1 La fonction **racine_deux** : `float -> float` qui renvoie une valeur approchée à ε près de $\sqrt{2}$, où ε est le paramètre transmis à cette fonction.

On admettra que la suite de réels définies par

$$u_0 = 1$$
$$u_{n+1} = \frac{u_n + 2/u_n}{2} \quad \forall n \in \mathbb{N}$$

converge vers $\sqrt{2}$ et que le terme u_n est une valeur approchée à ε près si $|u_n - u_{n+1}| < \varepsilon$.

La fonction **abs_float** : `float -> float` calcule la valeur absolue de son argument.

Question 2 La fonction **pgcd** : `int -> int -> int` qui calcule le pgcd des deux entiers passés en paramètre. (On rappelle que lorsque $b \neq 0$, $\text{pgcd}(a, b) = \text{pgcd}(b, a \bmod b)$, et que $\text{pgcd}(a, 0) = a$.)

Question 3 La fonction **longueur** : `'a list -> int` qui calcule la longueur de la liste passée en paramètre.

Question 4 La fonction **renverse** : `'a list -> 'a list` qui construit une liste dont les éléments sont ceux de la liste passée en paramètre dans l'ordre inverse.

Question 5 La fonction **map** : `('a -> 'b) -> 'a list -> 'b list` équivalente à **List.map**.