

Les Fonctions Récursives

1. Définitions de fonctions récursives.

Définition : Une fonction récursive est une fonction dont la définition fait appel à elle-même.

Syntaxe : **let rec** *nom* $p_1 \dots p_n = \text{expr} ;;$

Important : Le mot réservé **rec** est obligatoire pour indiquer qu'il s'agit d'une fonction récursive.

Remarque : Pour pouvoir traiter les cas de bases et le cas général, il est utile d'utiliser la structure if-then-else dans la partie *expr*.

L'exemple suivant montre comment définir, en CAML, la fonction récursive « factorielle » nommée *fact* :

```
# let rec fact n = if n = 0 then 1 else n * fact (n-1) ;;
```

```
fact : int -> int = <fun>
```

2. Ecrire en langage CAML les fonctions récursives suivantes :

- Fact : factoriel d'un entier n .
- Exp qui calcule la fonction : $\text{Exp}(x, y) = x^y$.
- Sigma : la somme des entiers de 0 à x .
- Sigma2 : la somme des entiers compris entre 2 entiers a et b.
- Fib : la fonction **Fibonacci** définie par :

$$\text{Fib}(n) = \begin{cases} 0 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ \text{Fib}(n-1) + \text{Fib}(n-2) & \text{sinon} \end{cases}$$
- pgcd : la fonction qui calcule le PGCD de 2 entiers, en utilisant l'algorithme d'Euclide :

$$\text{PGCD}(a,b) = \text{PGCD}(b,r), \text{ où } r \text{ est le reste de la division de } a \text{ par } b.$$

NB : Vous pourrez utiliser la fonction primitive (a mod b) qui renvoie le reste r.

- pgcd2 : la fonction qui calcule le PGCD de 2 entiers, en utilisant l'algorithme suivant :

$$\text{PGCD}(a, b) = \begin{cases} a & \text{si } a = b \\ \text{PGCD}(a-b, b) & \text{si } a > b \\ \text{PGCD}(a, b-a) & \text{si } b > a \end{cases}$$

- SumSerie : la fonction qui calcule la somme des n premiers termes de la série harmonique de la forme :

$$1 + 1/2 + 1/3 + \dots + 1/n \quad (\text{pour } n = 0, \text{ la somme est } 0).$$