

# TD 5 d'Introduction à la programmation

## Exercices sur la récursivité

### 1 La suite de *Fibonacci*

La suite de *Fibonacci* est définie de la manière suivante :

$$\begin{cases} u_0 = 1 \\ u_1 = 1 \\ \forall n > 1 \quad u_n = u_{n-1} + u_{n-2} \end{cases}$$

Écrire une fonction en *Caml* qui, étant donné un entier  $n$ , calcule la valeur de la suite de *Fibonacci* en  $n$ . Pour quelles valeurs de  $n$  est-ce que la fonction que vous avez écrite converge (aboutit à un résultat et s'arrête) ?

### 2 La suite factorielle

Définir la suite factorielle sous la forme d'une suite récurrente et en déduire une fonction en *Caml* qui, à partir d'un entier  $n$ , renvoie la valeur de  $n!$ .

### 3 Élever un nombre flottant à une puissance entière

Écrire en *Caml* une fonction qui, à partir d'un entier positif  $n$  et d'un flottant  $x$ , renvoie la valeur de  $x^n$ . Avant d'écrire cette fonction, élaborer un algorithme qui calcule cette valeur de façon récursive.

### 4 Quelques mots sur la redondance des calculs

1. Pour la fonction qui calcule la suite de *Fibonacci*, combien de fois la fonction *Fibonacci* est-elle appelée sur les entiers  $N - 1, N - 2, \dots, N - 5$ , lors du calcul de la valeur de la fonction pour un entier  $N$  ?
2. Soit la suite  $V_n$  définie de la façon suivante :

$$\begin{cases} V_0 = (1, 1) \\ V_1 = (1, 2) \\ \forall n > 1 \quad V_n = (u_n, u_{n+1}) \end{cases}$$

où  $(u_n)$  est la suite de *Fibonacci*. Exprimer  $V_{n+1}$  en fonction de  $V_n$ . On pourra utiliser les fonctions de projection *premier* et *second* qui à partir de  $(x, y)$  renvoient respectivement  $x$  ou  $y$ .

3. On cherche une nouvelle fonction qui calcule la suite de *Fibonacci* mais où, pendant le calcul de la valeur de cette suite pour un entier  $N$ , la fonction n'est appelée qu'une seule fois sur chacune des valeurs inférieures à  $N$ . En particulier, si  $(u_n)$  est la suite de *Fibonacci*, pour le calcul de  $u_n$ , il est inutile de calculer  $u_{n-1}$  **ET**  $u_{n-2}$  puisque  $u_{n-2}$  a déjà été déterminé pendant le calcul de  $u_{n-1}$ . Donc il est intéressant de pouvoir conserver certaines valeurs, au lieu de les recalculer. Pour cela, on commencera par écrire une fonction *FibboAux* qui à partir d'un entier  $n$  calcule la valeur de la suite  $V_n$ .
4. En déduire une fonction qui calcule la suite de *Fibonacci* en ne calculant chacun des termes de la suite qu'une seule fois.