Programmation fonctionnelle 2023-2024

Feuille 2 : Premières fonctions

Exercice 2.1 1. Écrire une fonction `test` qui prend en arguments trois entiers `x`, `y`, `z` et retourne `true` si `z` est la somme de `x` et `y`, et `false` sinon.

2. Utiliser la fonction `test` pour les valeurs 1, 2, 3, puis 2, 3, 4 des arguments `x`, `y`, `z`.

Exercice 2.2 Écrire une fonction `valeur_p4` qui prend en argument un entier x et retourne $3x^4 + 7x - 1$.

Exercice 2.3 Écrire une fonction `polynome3 a b c` qui prend en arguments trois entiers `a, b, c` et retourne la fonction polynôme $x \mapsto ax^2 + bx + c$. Donner des exemples d'appels de la fonction retournée par `polynome3`

Exercice 2.4 Soit la fonction `fact` définie comme suit :

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

Dessiner la pile des appels pour `fact 5`.

Exercice 2.5 Soient n, p deux entiers. On rappelle que $\text{pgcd}(n, 0) = n$ et que $\text{pgcd}(n, p) = \text{pgcd}(p, n \mod p)$. Écrire une fonction `pgcd` calculant le pgcd de deux entiers.

Exercice 2.6 La fonction d'Ackermann est un exemple de fonction à croissance **très** rapide. Elle est définie par

$$ack(m, n) = \begin{cases} n+1 & \text{si } m = 0 \\ ack(m-1, 1) & \text{si } m > 0 \text{ et } n = 0 \\ ack(m-1, ack(m, n-1)) & \text{sinon} \end{cases}$$

Écrire un programme `ack` calculant cette fonction. **Attention à ne pas la tester sur de trop grands entiers.**

Exercice 2.7 La suite de Fibonacci est définie par $u_0 = u_1 = 1$ et pour tout $n \geqslant 2$, $u_n = u_{n-1} + u_{n-2}$.

1. Écrire une fonction `fib` calculant le n-ème terme de cette suite.
2. Combien de sommes sont utilisées lors de l'appel `fib n` ?
3. En utilisant une fonction auxiliaire qui calcule le couple (u_n, u_{n+1}), améliorer `fib` pour que l'appel `fib n` n'utilise qu'un nombre linéaire de sommes.

Exercice 2.8 Supposément posé par les recruteurs d'**Amazon**.

- https://youtu.be/5o-kdjv7FD0
- Une personne monte un escalier à n marches.
- Elle monte à chaque pas soit une, soit deux, soit trois marches.
- De combien de façons peut-elle monter l'escalier ?

Écrire une fonction `stairs` qui prend en argument un entier n et calcule le nombre de façons de gravir un escalier de n marches sachant qu'on peut choisir de monter une, deux ou trois marches à chaque pas.

Exercice 2.9 1. Écrire une fonction `power` qui prend en argument deux entiers b et n avec $b \neq 0$, et qui calcule b^n.

2. Combien de multiplications sont effectuées lors de l'appel `power b n` ?

3. Peut-on en utiliser moins ?

Exercice 2.10 Écrire une fonction `iterate` qui prend en argument une fonction f et un entier k et qui renvoie la fonction $x \mapsto \underbrace{f(f(\cdots f}_{k \text{ fois}}(x)\cdots)$. Quel est le type de cette fonction ?