

Colle 04

Cette quatrième colle vous fera reconnaître du code OCaml, et écrire quelques fonctions simples en OCaml.

Fonction arithmétique

- Écrivez en OCaml une fonction `moyenne_six` qui prend trois entiers u, v, w, x, y, z (on verra plus tard les listes et tableaux) et renvoie leur moyenne comme un *nombre flottant* (pas comme un entier). Attention, on ne s'autorisera qu'au plus deux appels à la fonction de conversion de type `float_of_int`;
- Donnez à l'écrit la signature sous forme curryfiée, et sous forme décurryfiée. Quel est l'avantage d'une forme curryfiée ?

Algorithme d'Euclide pour le calcul du plus grand diviseur commun

Si $x < y$, $\text{PGCD}(x, y) = \text{PGCD}(y, x)$ pour avoir $x \geq y$ dans les cas suivants. On rappelle la relation $\text{PGCD}(x, y) = x$ si y divise x , et la relation de récurrence $\text{PGCD}(x, y) = \text{PGCD}(y, x \bmod y)$ si $x > y$, et le cas de base $\text{PGCD}(x, y) = x$ si $y = 0$. Écrivez une fonction *réursive* `PGCD` en OCaml, a deux arguments entiers x et y , renvoyant leur PGCD.

Illustrez avec plusieurs exemples bien choisis.

Test de primalité

Écrivez le pseudo-code d'une fonction qui teste si un nombre entier $n \geq 0$ donné en argument est premier ou non. Il faudra tester au plus \sqrt{n} diviseurs, et pas au plus $\mathcal{O}(n)$ diviseurs.

Bonus : écrire la fonction correspondante en OCaml.

QCM

Le(s)quel(s) des codes suivants sera une fonction OCaml valide ? (et il faudra pouvoir justifier) Corrigez les morceaux qui ne vont pas pour les lignes incorrectes.

- `def maximum x = fun y -> If x < y Then y Else x;;`
- `let minimum = fun x -> fun y -> if x < y then x else y;;`
- `def est_plus_petit (x, y) = (<) x y;;`
- `let est_plus_petit = if x < y then true else false;;`

Fonction sur un tableau

Au DS numéro 1, l'exercice 1 demandait de calculer la seconde plus grande valeur d'un tableau. Ici, on va se restreindre à calculer la plus grande valeur d'un tableau `tab` (de type homogène quelconque).

Écrivez une fonction OCaml qui accepte un `'a array` et renvoie son maximum de type `'a`. On supposera que le tableau est non vide. On rappelle la syntaxe `tab.(i)` pour lire la i -ème case de ce tableau, et `Array.length tab` donne sa longueur.