# Programmation Fonctionnelle en Scheme

**Examen de Rattrapage L1-I/MI**

*Université Nice Sophia-Antipolis, Juin 2017*

*LE FORMULAIRE FOURNI EST LE SEUL DOCUMENT AUTORISÉ. Tâchez de soigner l'indentation, la limpidité et l'exactitude du code. Répondez aux questions sur la copie d'examen qui vous est fournie, ne joignez aucune autre feuille ni intercalaire. N'hésitez pas à utiliser le résultat d'une question précédente même si vous ne l'avez pas traitée. Et ne perdez pas de temps...*

## 1 Styles Impératif et Fonctionnel [3 pts]

**a)** Sans faire référence à Python ou à Scheme, quel aspect linguistique essentiel distingue la programmation impérative de la programmation fonctionnelle ?

**b)** Peut-on coder en Scheme des *boucles* comme en Python malgré l'absence des mots-clés `while` et `for` ? Donnez un exemple simple de fonction itérative en Scheme.

## 2 Exécution d'une Fonction Récursive [2 pts]

Quelle est la valeur de l'expression `(foo 12 empty)` si l'on définit la fonction `foo` comme ci-dessous ?

```
(define (foo x L)
  (cond ((= x 1) L)
        ((= (modulo x 2) 0) (cons x (foo (/ x 2) L)))
        (else (foo (+ (* 3 x) 1) (cons x L)))))
```

## 3 Programmation sur les Nombres et les Listes [8 pts]

**a)** *Question préparatoire.* Programmez une fonction `(suppr-mult k L)` prenant un entier `k` et une liste d'entiers `L`, et retournant la liste obtenue en supprimant de `L` tous les multiples de `k`.

`(suppr-mult 3 '(4 6 2 1 9 12))` $\leadsto$ `(4 2 1)`

**b)** Programmez la fonction `(diviseurs n)` qui prend un entier $n \geq 2$, et retourne la liste croissante des **diviseurs** de $n$ appartenant à l'intervalle $[2, n]$.

`(diviseurs 42)` $\leadsto$ `(2 3 6 7 14 21 42)`, `(diviseurs 19)` $\leadsto$ `(19)`

**c)** Nous souhaitons maintenant obtenir la liste des **diviseurs premiers distincts** de $n$. Il nous suffit de supprimer les nombres *non premiers* de la liste `L` de tous les diviseurs, obtenue en b). En nous inspirant du *crible d'Eratosthène* étudié cette année en Python et en Scheme, il suffit de parcourir cette liste croissante `L` pour éliminer au fur et à mesure les multiples de chaque élément. Programmez par récurrence la fonction `(crible L)` prenant une liste `L` résultat de la fonction `diviseurs`, et retournant la liste obtenue en procédant à toutes ces éliminations successives.

```
(crible '(2 3 6 7 14 21 42)) ⇝ (2 3 7)
```

**d)** Déduisez de a) et c), en une ligne si possible, une fonction `(diviseurs-premiers n)` retournant la liste des diviseurs premiers distincts de l'entier `n` $\geq 2$.

```
(diviseurs-premiers 4840) ⇝ (2 5 11)        ; 4840 = 2^3 × 5^1 × 11^2
```

**e)** La *fonction indicatrice d'Euler* est définie pour un entier $n$ par $\varphi(n) = n(1 - \frac{1}{p_1}) \ldots (1 - \frac{1}{p_k})$ où $p_1, \ldots, p_k$ sont les diviseurs premiers distincts de $n$. Programmez la fonction `(phi n)` retournant $\varphi(n)$. Exemple : $\varphi(12) = 4$ puisque les diviseurs premiers de 12 sont 2 et 3, et que $12(1-\frac{1}{2})(1-\frac{1}{3}) = 4$. En utilisant un schéma d'ordre supérieur, complétez la ligne ci-dessous en écrivant l'expression `<2>` sur votre copie.

```
(define (phi n)                                  ; n(1 - 1/p1)...(1 - 1/pk)
  (* n <2>))
```

## 4 Recherche séquentielle dans une liste non triée [4 pts]

*Les deux questions sont indépendantes.*

**a)** Programmez une fonction `(compte L x)` retournant le nombre d'apparitions de l'élément `x` dans la liste `L`. Exemple :

```
(compte '(a b b a b c) 'b)   ⇝ 3
(compte '(a b b a b c) 'd)   ⇝ 0
```

**b)** Programmez la fonction `(coupe L x)` prenant une liste `L` et une donnée `x`, et retournant une liste à deux éléments `(L1 L2)` où `L1` est la sous-liste des éléments de `L` situés avant la première apparition de `x`, et `L2` la sous-liste des éléments de `L` à partir de `x`. Dans le cas où `x` n'est pas un élément de `L`, la fonction retournera `#f`.

```
(coupe '(le gros chien qui aboie) 'qui)   ⇝ ((le gros chien) (qui aboie))
(coupe '(le gros chien que je voie) 'qui) ⇝ #f
```

## 5 Recherche dichotomique dans un intervalle réel [3 pts]

Soit $f : [a, b] \to \mathbb{R}$ une fonction continue sur $[a, b]$ telle que $f(a) < 0 < f(b)$. Vous avez vu en Analyse qu'il existe alors au moins un nombre réel $x \in [a, b]$ sur lequel $f(x) = 0$. Il peut y en avoir plusieurs, le but est d'en trouver un, avec une précision de $h$, en utilisant la technique suivante. Soit $m$ le milieu de $[a, b]$. Si la largeur de $[a, b]$ est inférieure à $h$, alors $m$ est solution. Sinon, montrez qu'en étudiant les trois cas $f(m) < 0$, $f(m) > 0$ et $f(m) = 0$, vous pouvez couper en deux l'intervalle de recherche et converger ainsi vers une solution. Programmez la fonction `(une-racine f a b h)` retournant avec une précision de $h$ un point $x \in [a, b]$ tel que $f(x) = 0$.
Exemple avec la fonction $x \mapsto x^2 - 2$ telle que $f(0) < 0 < f(2)$. La solution sur $[0, 2]$ est $\sqrt{2}$ :

```
(une-racine (lambda (x) (- (* x x) 2)) #i0 #i2 #i0.001) ⇝ #i1.41455078125
```