

## TD numéro 2

### 1 Évaluation d'expressions en Scheme

Donner la valeur retournée par les expressions ci-dessous :

- (+ 2 6)
- (+ (\* 2 3 5) (- 6 8))
- '(+ 2 6)
- '(1 2 3)
- (1 2 3)

On suppose que les définitions suivantes ont été faites dans l'ordre donné :

(define moineau 5)

(define condor 435)

(define rapace 'condor)

(define oiseau condor)

Trouver les résultats des évaluations suivantes:

- baleine
- 'baleine
- moineau
- 'moineau
- rapace
- (+ oiseau moineau)
- (+ rapace condor)
- (+ (eval rapace) condor)
- (and (> 21 45) (= 3 (/ 12 4)))
- (and (> 21 45) (= 3 (/ 12 0)))
- (and (= 3 (/ 12 0)) (> 21 45))
- (boolean? (number? 3))
- (boolean? (number? "abc"))
- Soit expr une expression booléenne, évaluer :
  - (not (or expr true))
  - (not (and (or expr false) (not expr)))

### 2 Premières fonctions en Scheme

Écrire:

- une fonction qui renvoie le double d'un nombre passé en argument ;
- une fonction qui renvoie la moyenne de deux nombres passés en argument ;
- une fonction qui renvoie le signe d'un nombre ;
- une fonction qui renvoie la mention pour une note donnée ;
- une fonction qui calcule le n<sup>ième</sup> terme de la suite de Fibonacci ( $u_n = u_{n-1} + u_{n-2}$ ,  $u_0 = 1$ ,  $u_1 = 1$ )

Donner la spécification de la fonction mystère ci-dessous :

```
(define mystere
  (lambda (x)
    (= 0 (modulo x 2))))
```