

TD numéro 1 : débuts en Scheme

1 Évaluation d'expressions en Scheme

Donner la valeur retournée par les expressions Scheme écrites ci-dessous :

- (+ 2 6)
- (+ (* 2 3 5) (- 6 8))
- '(+ 2 6)
- (and (> 21 45) (= 3 (/ 12 4)))
- (and (> 21 45) (= 3 (/ 12 0)))
- (and (= 3 (/ 12 0)) (> 21 45))
- '(1 2 3)
- (1 2 3)
- (= 0 (modulo 17 2))
- (boolean? (number? 3))
- (boolean? (number? "abc"))

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)

Soit expr une expression booléenne, évaluer :

- (not (or expr true))
- (not (and (or expr false) (not expr)))

2 Premières fonctions en Scheme

Définir en Scheme :

- 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 un booléen spécifiant si le nombre passé en argument est positif.
- une fonction qui renvoie la mention pour une note donnée.
- une fonction récursive qui renvoie la somme des n premiers entiers.
- une fonction qui calcule le $n^{\text{ième}}$ terme de la suite de Fibonacci. Nous rappelons que cette suite est définie par : $u_0 = 1, u_1 = 1, u_n = u_{n-1} + u_{n-2}$.

3 Fonction mystère en Scheme

- Donner la spécification de la fonction Scheme ci-dessous :

```
(define mystere
  (lambda (n)
    (if (= n 0)
        0
        (if (= (modulo n 2) 0)
            (+ n (mystere (- n 1)))
            (mystere (- n 1))))))
```