

TD numéro 2 : listes

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

- | | |
|--------------------------------------|------------------------------|
| • (car (cdr '(a b c d))) | • (cons '(a b) '(c d)) |
| • (car (cdr '(abc d))) | • (cons 'a (cons 'b '(c d))) |
| • (cdr '(a (b c d))) | • (cons '(a b) 'c) |
| • (cdr '((a b c d))) | • (list? (+ 2 3)) |
| • (cdr (car (cdr '(a (b c) (d e))))) | • (list? '(+ 2 3)) |

- Définir en Scheme :

- une fonction qui retourne le second élément d'une liste d'au moins deux éléments ;
- une fonction qui retourne vrai si et seulement si une liste n'a qu'un seul élément ;
- une fonction qui calcule la longueur d'une liste (i.e. son nombre d'éléments).
- une fonction qui prend une liste et un élément et retourne vrai si et seulement si l'élément appartient à la liste ;
- une fonction qui retourne le n^{ième} élément d'une liste ;
- une fonction qui insère un élément dans une liste après le i^{ème} élément ;
- une fonction qui retourne le dernier élément d'une liste non vide.

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

```
(define mystere
  (lambda (x l)
    (cond ((null? l) 0)
          ((eq? x (car l)) (+ 1 (mystere x (cdr l))))
          (else (mystere x (cdr l))))))
```