ENSEIRB-Matmeca
2019-2020

1ère année - Filière Info
PG104

TD n°4 - Paires pointées, Listes

Exercice 1: Évaluation

Évaluer les expressions suivantes après en avoir deviné le résultat :

1. `(cons '(A B C) '(1 2 3))`
2. `(append '(A B C) '((1 2) 3))`
3. `(last '((A 1) (B 2) (C 3)))`
4. `(drop-right '((A 1) (B 2) (C 3)) 1)`
5. `(car '((A (B C)) D (E F)))`
6. `(cdr '((A (B C)) D (E F)))`
7. `(caddr'((A (B C)) D (E F)))`
8. `(cons 'NOBODY (cons 'IS '(PERFECT)))`
9. `(list (add1 2) (sub1 5) 6)`
10. `(cdr '(a b))`
11. `(cdr '(a . b))`
12. `'(a . (b . (c . ())))`
13. `'(a . (b . (c . d)))`
14. `(assoc 'bleu '([rouge . red] [vert . green] [bleu . blue] [jaune . pink]))`

Remarque : ne pas hésiter à chercher dans la documentation toute fonction inconnue.

Exercice 2: Car et Cdr

Donner les combinaisons de `car` et `cdr` nécessaires pour remplacer le signe "?" dans chacun des cas suivants afin que :

1. `(? '(A B C D))` ait pour résultat D
2. `(? '((A (B C)) E))` ait pour résultat C
3. `(? '(((FLUTE) ENCORE) UNE))` ait pour résultat FLUTE
4. `(? '(((FLUTE) ENCORE) UNE))` ait pour résultat ENCORE

Exercice 3: Manipulation de listes

1. Utiliser `car`, `cdr` et `cons` pour inverser une liste à 3 éléments.
2. Écrire une autre fonction d'inversion, qui utilise trois fonctions auxiliaires : `first`, `second` et `third`, permettant d'inverser une liste à trois éléments.
3. Construire la liste `'(1 2 3 4 5)` à partir de `'(1 2)`, `'((3 4))` et `5`, avec `cons` puis avec `append`.

Exercice 4: Valeur absolue

Écrire une fonction récursive `list-abs-recursif` (sans utiliser un équivalent de `map`) qui accepte en entrée une liste de nombres et qui retourne la liste constituée de la valeur absolue de chaque élément de la liste fournie en paramètre.

Exercice 5: Construction de liste

1. Écrire une fonction `repeat` qui construit une liste contenant n fois l'élément e.
2. Obtenir la liste `(glou glou glou glou glou)`.

Exercice 6: Questions de tris

1. Écrire une fonction `insert(x,l)` qui insère un élément x dans une liste triée l.
2. Écrire une fonction `sort-numbers(l)` qui trie une liste de nombres l en utilisant l'algorithme de tri par insertion.
3. Écrire une fonction `sort-symbols(l)` qui modifie la fonction précédente pour trier une liste de symboles l selon l'ordre lexicographique (par exemple on pourra utiliser le prédicat `symbol<?`).
4. Pour aller plus loin... Écrire une fonction de tri utilisant l'algorithme de tri fusion.

Exercice 7: Échange d'éléments

Écrire une fonction `swap-first-last(l)` qui "échange" le premier et le dernier élément d'une liste (*Remarque :* la fonction `drop-right` peut aider).

La tester avec la liste `'(YOU CANT BUY LOVE)`.

Exercice 8: Rotation de liste

1. Écrire une fonction `rotate-left(l)` qui fait une rotation circulaire vers la gauche des éléments d'une liste, et la tester.
2. Écrire une fonction `rotate-right(l)` qui fait une rotation circulaire vers la droite des éléments d'une liste et la tester.