

UE INF353

Programmation 3

Programmation Fonctionnelle et Symbolique

Devoir surveillé No 1

Tous documents autorisés.

Mercredi 22 Octobre 2008

Durée : 1h10.

Le barème est donné à titre **indicatif**.

Chacun des trois exercices doit être **rédigé** sur une feuille **séparée** sur laquelle doivent être inscrits votre nom et votre numéro de groupe. Vous devez rendre exactement **une feuille** (ou copie), éventuellement blanche, **par exercice**.

*Dans les exercices 2 et 3 il est **obligatoire** de programmer dans un style **fonctionnel**, c'est-à-dire sans utiliser ni **setf** ni **boucle**.*

Exercice 1 (6pts)

Évaluer les expressions suivantes :

1. `(cons '(1) '(2 3 4))`
2. `(cons nil nil)`
3. `(list nil nil)`
4. `(append nil nil)`
5. `'(+ 1 2 3 4)`
6. `(last '(1 2 3 4))`
7. `(cons 3 (butlast '(1 2 3)))`
8. `(member 6 '(0 3 6 9 12))`
9. `(and (zerop (mod 9 3)) (mod 9 3))`
10. `(mapcar #'cons '(1 3 5 7) '(2 4 6 8))`
11. `(every (lambda (x) (zerop (mod x 3))) '(0 3 6 9 12))`
12. `(equal '(1 (2 3)) (list 1 (list 2 3)))`

Exercice 2 (9pts)

Soit n un entier positif. À tout entier $i < n$ on associe l'ensemble $D_n(i)$ composé des éléments inférieurs ou égaux à n et de la forme

$$i2^p + q \quad \text{pour } p \geq 0 \text{ et } 0 \leq q < 2^p$$

Ainsi on a :

$$D_n(i) = \{i, 2i, 2i + 1, 4i, 4i + 1, 4i + 2, 4i + 3, \dots, 2^p i + q, \dots\}$$

Par exemple pour $n = 46$ on a

- $D_{46}(9) = \{9, 18, 19, 36, 37, 38, 39\}$
- $D_{46}(11) = \{11, 22, 23, 44, 45, 46\}$

1. Calculer les ensembles $D_{53}(13)$ et $D_{53}(20)$.
2. Écrire une fonction `D (n i)` qui retourne l'ensemble $D_n(i)$ sous forme de liste.
 N.B. L'**ordre** des éléments dans la liste est **sans importance** étant donné qu'il s'agit d'un ensemble.
 Exemple : `(D 46 9)` peut retourner `(9 18 36 37 19 38 39)` tout autant que `(9 18 19 36 37 38 39)`, `(39 38 37 36 19 18 9)` ou n'importe quelle autre permutation de ces nombres.
3. Écrire une fonction `D-n (n)` qui retourne la fonction $D_n : i \mapsto D_n(i)$.
4. Écrire un exemple d'appel d'une fonction retournée par `D-n` ainsi que la valeur retournée par cet appel.

Exercice 3 (5pts)

1. Écrire une fonction `count-pairs tree` qui compte le nombre de paires constituant l'arbre `tree`.
2. Que retourne l'appel `(count-pairs '((1 . 2) (3 . 4) (5 . 6)))`?

Exemples :

```
CL-USER> (count-pairs 1)
0
CL-USER> (count-pairs nil)
0
CL-USER> (count-pairs '(1 . 2))
1
CL-USER> (count-pairs '((1 . 2) . (3 . 4)))
3
CL-USER> (count-pairs '(((1 . 2) . 3) . ((4 . 5) . (6 . 7))))
6
CL-USER> (count-pairs '(1 2 3 4))
4
```

FIN