ANNÉE UNIVERSITAIRE 2014/2015
1ÈRE SESSION

Parcours : Licence Informatique - Semestre 5
Épreuve : Programmation 3
Date : janvier 2015
Durée : 1h30
Documents : non autorisés
Épreuve de Mme Irène Durand

Le barème est donné à titre **indicatif**. Le sujet comporte 2 pages.

Exercice 1 *(2pts)*

Évaluer les expressions suivantes :

1. `(cons 1 (cons 2 nil))`
2. `(list 1 (list 2 nil))`
3. `(append '(1 2) nil '(3 4))`
4. `(append '(1 2) 3)`

Exercice 2 *(3pts)*

5. Écrire une fonction `compte-occ (atom l)` qui compte les occurrences d'un atome `atom` dans une liste *généralisée*[1] `l`.
   Exemples :

```
CL-USER> (compte-occ 1 '(1 (2 3 1) (1 (3 1) 1)))
5
CL-USER> (compte-occ 3 '(1 (2 3 1) (1 (3 1) 1)))
2
```

Exercice 3 *(7pts)*

6. Définir la fonction `compose (f g)` qui s'applique à deux fonctions d'une variable et retourne la fonction composée ($g \circ f$), c'est-à-dire la fonction telle que $\forall x, (g \circ f)(x) = g(f(x))$.
7. Écrire une expression qui fait appel à la fonction retournée par un appel à `compose` et sa valeur de retour.

Soit la fonction `compose-n (f n)` qui retourne la fonction $f^n = f \circ \ldots \circ f$. On rappelle que f^0 est la fonction identité.

8. Écrire une version **récursive** de la fonction `compose-n (f n)`.
9. Votre fonction est-elle récursive terminale ? Justifiez votre réponse.
10. Écrire une version **itérative** de la fonction `compose-n (f n)`.

1. Une liste généralisée est une liste pouvant contenir des sous-listes.

Exercice 4 *(8pts)*
On dispose d'un *vecteur* (tableau à une dimension) de taille arbitraire tel que la case `i` contient la chaîne de caractères correspondant à l'écriture du nombre `i` en toutes lettres. À partir de ce vecteur, on souhaite construire une *table de hachage* permettant de traduire un chiffre écrit en toutes lettres (chaîne de caractères) en sa valeur numérique (entier).

11. Écrire une fonction `make-number-table (vector)` qui retourne une telle table.
    Exemples :

```
CL-USER> *vecteur-nombres*
#("zero" "un" "deux" "trois" "quatre" "cinq" "six" "sept" "huit" "neuf" "dix")
CL-USER> (setf *table-nombres*  (make-number-table *vecteur-nombres*))
#<HASH-TABLE :TEST EQUAL :COUNT 11 {1003367E71}>
CL-USER> (gethash "trois" *table-nombres*)
3
T
CL-USER> (gethash "cent" *table-nombres*)
NIL
NIL
```

On dispose d'une table de hachage traduisant des nombres entiers écrits en toutes lettres en leur valeur numérique. On suppose que si la table contient n nombres, elle contient les nombres de 0 à $n-1$.

12. Écrire une fonction `vector-from-table (table)` qui à partir d'une telle table reconstruit le vecteur d'origine.
    Exemple :

```
CL-USER> (vector-from-table *table-nombres*)
#("zero" "un" "deux" "trois" "quatre" "cinq" "six" "sept" "huit" "neuf" "dix")
```

FIN