



ANNÉE UNIVERSITAIRE 2011/2012
1ÈRE SESSION



Parcours : CSB5
Code UE : INF353
Épreuve : Programmation 3
Date : Vendredi 06 janvier 2012
Heure :
Durée : 1h30
Épreuve de : Mme Irène Durand

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

Le sujet comporte 3 pages plus une annexe.

Exercice 1 (2pts)

Évaluer les expressions suivantes :

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

Exercice 2 (3pts)

Soit le programme :

```
(let ((cpt 0))
  (defun cpt-next ()
    (setf cpt (1+ cpt)))
  (defun cpt-set (&optional (n 0))
    (setf cpt n)))
```

Compléter le scénario suivant :

```
CL-USER> (cpt-next)                ;; Réponse 1
CL-USER> (cpt-next)                ;; Réponse 2
CL-USER> (cpt-set)                 ;; Réponse 3
CL-USER> (cpt-next)                ;; Réponse 4
CL-USER> (cpt-set 10)              ;; Réponse 5
CL-USER> (cpt-next)                ;; Réponse 6
```

Exercice 3 (5pts)

1. Écrire une fonction `serie (f n)`, où `n` est un entier et `f` une fonction numérique, qui calcule

$$\sum_{i=0}^n (-1)^i f(2i+1) = f(1) - f(3) + f(5) - \dots + (-1)^n f(2n+1)$$

2. Que retourne l'appel (`serie #'1+ 2`) ?
3. Que retourne l'appel (`serie #' / 2`) ?

Sachant que le développement en série suivant calcule une valeur approchée de $\pi/4$:

$$\pi/4 = 1 - \frac{1}{3} + \frac{1}{5} - \dots = \sum_{n=0}^{\infty} (-1)^n \frac{1}{2n+1}$$

4. Donner une expression donnant une valeur approchée de π .

Exercice 4 (10pts)

Soient les trois fonctions `mystere1`, `mystere2` et `mystere3` suivantes :

```
(defun mystere1 (l f)
  (if (endp l)
      '()
      (if (funcall f (car l))
          (cons (car l) (mystere1 (cdr l) f))
          (mystere1 (cdr l) f))))

(defun mystere2 (l f)
  (labels ((aux (l lres)
            (if (endp l)
                lres
                (aux (cdr l)
                    (if (funcall f (car l))
                        (append lres (list (car l)))
                        lres))))))
    (aux l '())))

(defun mystere3 (l f)
  (labels ((aux (l lres)
            (if (endp l)
                (nreverse lres)
                (aux (cdr l)
                    (if (funcall f (car l))
                        (cons (car l) lres)
                        lres))))))
    (aux l '())))
```

1. Que retourne l'appel (`mystere1 '(1 2 3 4 5) #'evenp`) ?

2. Que retourne l'appel (`mystere2 '(1 2 3 4 5) #'evenp`) ?
3. Que retourne l'appel (`mystere3 '(1 2 3 4 5) #'evenp`) ?
4. Donner un exemple d'appel à `mystere3` ainsi que la valeur de retour de cet appel. Cet appel doit être **non trivial**, c'est-à-dire que 1 doit être différent de `NIL`.
5. Les fonctions `mystere1`, `mystere2` et `mystere3` sont-elles équivalentes (même résultats pour mêmes arguments) ?
6. Décrire en une phrase ce que calcule la fonction `mystere3` (ce qu'elle prend en entrée et ce qu'elle retourne en sortie).
7. Parmi ces trois fonctions, lesquelles sont *récur­sives terminales* ?

Soient les trois appels présentés Figure 1 dans l'Annexe.

8. Expliquer la différence de comportement entre ces trois appels.
9. Laquelle de ces trois fonctions est la plus efficace ?
10. Écrire une version itérative de la fonction `mystere3`.

FIN

Annexe

```
CL-USER> (defparameter *l* (make-list 50000 :initial-element 0))
*L*
CL-USER> (time (defparameter *r* (mystere1 *l* #'evenp)))
Evaluation took:
  0.003 seconds of real time
  0.000000 seconds of total run time (0.000000 user, 0.000000 system)
  0.00% CPU
  7,748,816 processor cycles
  798,720 bytes consed

*R*
CL-USER> (time (defparameter *r* (mystere2 *l* #'evenp)))
Evaluation took:
  37.101 seconds of real time
  36.830000 seconds of total run time (32.550000 user, 4.280000 system)
  [ Run times consist of 4.260 seconds GC time, and 32.570 seconds non-GC time. ]
  99.27% CPU
  98,685,763,984 processor cycles
  20,002,730,976 bytes consed

*R*
CL-USER> (time (defparameter *r* (mystere3 *l* #'evenp)))
Evaluation took:
  0.002 seconds of real time
  0.000000 seconds of total run time (0.000000 user, 0.000000 system)
  0.00% CPU
  3,369,376 processor cycles
  798,720 bytes consed

*R*
```

FIG. 1 – Exemple d'exécution