ENSEIRB-Matmeca – Filière informatique première année 2016/2017

Examen de programmation fonctionnelle

Durée 2h, documents autorisés

1. **[2 pts]** Expliquer en quoi consiste la propriété de transparence référentielle. Donner deux exemples de fonctions en C ou en Scheme, l'un respectant la transparence référentielle et l'autre non en expliquant pourquoi dans les deux cas.

2. **[2 pts]** Expliquer en quoi consiste la β-réduction et le rôle qu'elle joue dans l'évaluation des expressions en scheme. Donner deux exemples de β-réductions de l'expression suivante : `(/ (* (sqr (+ 1 3)) (+ 1 3)) (- 10 2))` Quelle stratégie d'évaluation est utilisée dans le langage scheme?

3. **[3 pts]** Compléter les expressions suivantes de façon à ce qu'elles soient correctes et qu'elles produisent le résultat demandé, en remplaçant le point d'interrogation par une expression.

```
>  (apply ? '(1 2 3 4 5))
15

> (map * ? '(1 2 3))
'(1 2 3)

> (foldr  ?  ? '(1 2 3 4 5))
'(1 2 3 4 5 6 7 8)

> (?  list '(1 2 3) '(2 3 4))
'((1 2) (2 3) (3 4))

> (? * 1 '(1 2 3 4))
24
```

4. **[4 pts]** Au vu des exemples vus en cours et en TD, et compte tenu des fonctions présentes dans la bibliothèque, critiquer l'écriture des fonctions suivantes.

```
(define (f x y z)
  (if ((zero? x) y)
      ((positive? x) z)
      (else x)))

(define (filtrer predicat liste)
  (if (predicat (car liste)) (cons (car liste) (filtrer (cdr liste)))
      (filtrer predicat (cdr liste))))

(define (valeur-absolue x)
  (if (< x 0)
      (- 0 x)
      x))

(define (racine1 a b c)
  (define (discriminant a b c)
    (- (* b b) (* 4 a c)))
  (if (> (discriminant a b c) 0)
      (/ (- (- b) (sqrt (discriminant a b c)))
```

```
      (* 2 a))
    'negative-discriminant))
```

5. **[5 pts]** Soit la fonction suivante :

```
(define (mystery l1 l2)
  (cond ((null? l1) '())
        (else (cons ((car l1) (car l2)) (mystery (cdr l1) (cdr l2))))))
```

   - Quel est le résultat de l'expression suivante :

     ```
     (mystery (list add1 sub1) '(1 2))
     ```

   - Proposer une autre version de cette fonction en utilisant la bibliothèque et permettant d'éviter l'écriture explicite de la boucle récursive.
   - Que fait la fonction suivante? Donner un exemple d'appel ainsi que le résultat attendu.

     ```
     (define (extend f . fargs)
       (lambda (x)
         (apply f (map (lambda (g) (g x)) fargs))))
     ```

6. **[4 pts]** Décrire les résultats des évaluations des expressions associées aux définitions des symboles suivants.

```
(define sqr1+ (compose sqr add1))

(define apply+ (curry apply +))

(define map* (curry map *))
```

On donne la définition fonctionnelle suivante du produit vectoriel

```
(define produit-vectoriel (compose apply+ map*))

> (produit-vectoriel '(1 2 3) '(2 3 4))
20
```

Sur le même modèle, écrire les fonctions `norme` et `rms` définies par :

$$norme(x) = \sqrt{\sum_{i=1}^{n} x_i^2}$$

$$rms(x) = \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}$$

où x est un vecteur de taille n.