



### Devoir surveillé de Programmation Fonctionnelle

Documents non autorisés

#### Exercice 1 (4pts)

Evaluer les expressions suivantes et donner les résultats.

1. (cdadr '((1) (2 3)))
2. (caddr '(1 2 3))
3. (append '(1) '(2 3))
4. (append nil '(1 2) nil)
5. (member 3 '(1 3 5 9 12))
6. (mapcar #'\* '(1 3 5) '(2 4 6) '(1 1 1))
7. (some (lambda (x) (and (zerop (mod x 3)) x)) '(1 2 3 4))
8. (mapcan (lambda (x) (if (= x 3) '() (list x))) '(1 3 5 4 6))

#### Exercice 2 (12pts)

Soient les deux fonctions anonyme1 et anonyme2 définies ci-dessous :

```
(defun anonyme (n)
  (if (zerop n)
      '()
      (cons n (anonyme (- n 1)))))
```

```
(defun anonyme1 (n)
  (reverse (anonyme n)))
```

```
(defun anonyme2 (n)
  (if (zerop n)
      '()
      (append (anonyme2 (- n 1)) (list n))))
```

1. Que retourne l'appel (anonyme1 6) ?
2. Que retourne l'appel (anonyme2 6) ?
3. Les fonctions anonyme1 et anonyme2 sont-elles équivalentes (mêmes effets pour mêmes arguments) ?
4. Que fait la fonction anonyme1 ?
5. Ecrire une fonction app-f1 (f L) qui retourne la liste des valeurs de la fonction f sur les éléments de L. Exemple : (app-f1 #'sqrt '(1 4 9 25)) → (1 2 3 5).
6. Ecrire une fonction app-f2 (f n) qui retourne la liste des valeurs de la fonction f sur les nombres allant de 1 à n. Exemple : (app-f2 #'evenp 5) → (NIL T NIL T NIL).

#### Exercice 3 (4pts)

Définir les fonctions suivantes :

1. avantDernierDe (L), qui retourne l'avant dernier élément de L.
2. different (L1 L2), qui retourne une liste constituée des éléments de L1 qui ne sont pas dans L2.

Bonne chance