Université de La Réunion — Lundi 31 mai 1999

DEUG 1^{ere} année – Tronc A – MIAS 1 et MASS 1

UE4 – Programmation fonctionnelle – Examen terminal **3**

Durée : 2 heures – Aucun document n'est autorisé.

Répondre uniquement dans les cadres prévus à cet effet

Nom :	No dossier :
Prénom(s) :	Signature :
Date de naissance :	

Exercice 3 : Ordre supérieur *(5 points)*

(3 pts) 3.a Donnez le résultat de l'évaluation des expressions suivantes :

```
(apply append '((1) () (2 a) (3 (u 2))) )
⟹
(map cdr '((1) (2 a) (3 (u 2))) )
⟹
(map car '(a (b)) )
⟹
(map (lambda (n) (and (even? n) (< 0 n))) '(-10 9 4 0 2))
⟹
((lambda (a b c) (- (* b b) (* 4 a c))) 1 2 1)
⟹
(apply + (map (lambda (x) 1) '(1 (2 3) 4 (5 6)) ))
⟹
```

(1 pt) 3.b Écrivez, en utilisant `apply` et `map`, une fonction non récursive `retirer` qui prend en argument un élément e et une liste plate l et qui retire de l toutes les occurrences de e.
Exemple : `(retirer 2 '(1 2 3 4 2) )` rend `(1 3 4)`.
On donne comme indice l'expression `(append '(1) '() '(3) '(4) '() )`.

(1 pt) 3.c En utilisant `apply` et `map`, écrivez une fonction non récursive `applatir` qui applatit une liste l en s'arrêtant au premier niveau.
Exemple : `(applatir '((1) 2) )` renvoie `(1 2)` et `(applatir '((1 (2) 3) 4) )` renvoie `(1 (2) 3 4)`.
On donne comme indice les expressions `(append '(1) '(2) )` et `(append '(1 (2) 3) '(4) )`.