

CADRE  
RÉSERVÉ AU  
CORRECTEUR

Note :  
/ 20

**Nom :** **Prénom :**  
**Numéro d'étudiant-e :**

0 0,5 1

**Premier exercice (2 points)**

Donner les résultats de l'évaluation par l'interpréteur Scheme des expressions suivantes :

`(cdr (append '(a b c) (list 'd '(e))))` → .....

`(list (car '(a b c)) (cadr '(a b c)))` → .....

**Deuxième exercice (4 points)**

On définit les listes suivantes :

`(define L1 '(a b c))`

`(define L2 '((d e) (f g)))`

Donner les expressions Scheme utilisant L1 et L2 et permettant d'obtenir les résultats suivants :

0 1 2

..... → (a f g)

..... → (b d)

**Comp. C1 :**  
/ 6

**Troisième exercice (7 points)**

Définir une fonction en Scheme qui permute 2 à 2 les éléments d'une liste.

`(permute '(a b c d e f))` → (b a d c f e)

`(permute '(a b c d e))` → (b a d c e)

Entête

0 1 2

Cas d'arrêt

0 0,5 1

Appel récursif

0 1 2

Const° résultat

0 1 2

**Comp. C2 :**  
/ 7

### Quatrième exercice (7 points)

Définir une fonction en Scheme qui retourne le nombre de nombres et le nombre de symboles contenus dans une liste plate. Attention à ne parcourir la liste qu'une seule fois.

(compte '(1 a b 2 d e f 4 3 5 g)) → (5 6)

*Rappel* : les fonctions booléennes `number?` et `symbol?` sont prédéfinies.

#### Maitrise en remontant :

**oui**      **non**  
     

Entête  
0   0,5   1  
     

Cas d'arrêt  
0   0,5   1  
     

Let/Comp. C3  
0   1,5   3  
     

Const° résultat  
0   1   2  
     

#### Maitrise en descendant :

**oui**      **non**  
     

Fonct° princ<sup>ale</sup>  
0   1   2  
     

Fonct° annexe  
0   0,5   1  
     

Cas d'arrêt  
0   1   2  
     

Appel récursif  
0   1   2  
     

Comp. C4 :