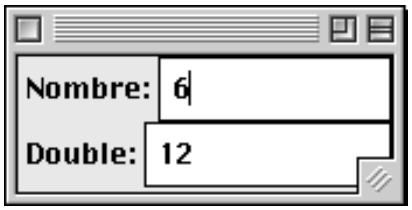
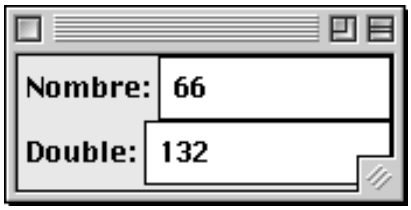
Informatique DEUG MIAS/MASS 1ère année $MI_{11}T_2$ Année 2002/2003 Juin

Durée 2 heures. Seule la petite documentation Scheme est autorisée. Les questions sont indépendantes.

Question 1
Evaluer :
a- (caddr '(a b c d))
b- (cdar '((1 2) (3 4)))
c- (cdr '(1 (2 (3))))
d- (null? (cddar '((a b) (c))))

Question 2
On souhaite construire l'interface ci-après. L'action de la deuxième zone texte est void. Dans la première zone texte, chaque saisie d'un chiffre déclenche l'action associée, qui consiste à afficher dans l'autre zone texte le nombre qu'elle contient multiplié par 2.





a- En utilisant string->number, faire (defaut t) qui traduit en nombre le contenu d'un objet text-field% lorsque celui-ci est non vide, et qui donne 0 si l'objet contient la chaîne vide "".
b- Construire l'interface.

Question 3
Evaluer :
a- (map (lambda (x) (- 10 x)) '(1 3 5 7 9))
b- (map (lambda (x y) (+ x y)) '(1 2 3) '(4 5 6))
c- (map (lambda (x) (if (even? x) 0 x)) '(1 2 3 4 5))

Question 4
Faire en *récursion terminale* (u u0 n) calculant la suite définie par u_0 quelconque et pour n>0 :
$u_n = \frac{1}{3}(2u_{n-1} + \frac{8}{{u_{n-1}}^2})$.

Question 5
Dans la divination *sikidy* pratiquée à Madagascar, on utilise des figures formées avec des graines sur le sol. À partir de huit figures initiales (*tale*, *maly*, etc.), on obtient huit nouvelles figures (*fahasivy*, *ombiasy*, etc.) en combinant entre elles des figures déjà placées. En Scheme, les huit figures initiales sont représentées par des listes contenant leur numéro de 1 à 8 : c1=(1), c2=(2) etc., et les autres par des listes contenant des chiffres de 1 à 8 ordonnés croissant, sans répétition : c9=(7 8), c10=(5 6 7 8), c11=(5 6), c12=(1 2 3 4 5 6 7 8), c13=(3 4), c14=(1 2 3 4), c15=(1 2), c16=(2 3 4 5 6 7 8).
a- Pour combiner deux figures, on cumule les numéros contenus dans leurs listes respectives, sauf les numéros communs qui sont supprimés. Faire en récursion non-terminale (xor ci cj) qui réalise la combinaison des figures. Exemple : (xor c11 c12) -> (1 2 3 4 7 8)
b- La divination consiste à combiner des figures existantes (par exemple, combiner c11 et c12 permet de dire si le mercredi est un jour propice ou non). Faire une fonction (enumere l) qui compte le nombre de figures différentes que l'on peut obtenir en combinant deux à deux les huit figures c9 à c16. Exemple : (enumere (list c9 c10 c11 c12 c13 c14 c15 c16)) -> 23
On fera une récursion terminale avec une variable res mémorisant les figures rencontrées.

EXAMEN INFORMATIQUE MIAS/MASS FORMULAIRE DE REPONSE Juin 2003

Numéro de place :

Chaque candidat écrit son nom dans le coin de la copie qu'il cachera par collage *après* l'émargement. Le numéro de place doit figurer sur chaque copie ou intercalaire.

1a c

1b (2)

1c ((2 (3)))

1d #t

2a
```
(define (defaut t)
  (let ((x (send t get-value)))
    (if (equal? x "") 0 (string->number x))))
```

2b
```
(define f (make-object frame% ""))
(send f show #t)
(define (action o e)
  (send t2 set-value (number->string (* 2 (defaut t1)))))
(define t1 (make-object text-field% "Nombre:" f action))
(define t2 (make-object text-field% "Double:" f void))
```

3a (9 7 5 3 1)

3b (5 7 9)

3c (1 0 3 0 5)

4
```
(define (u n u0)
  (cond ((= n 0) u0)
        (else (u (- n 1) (/ (+ (* 2 u0) (/ 8. (* u0 u0))) 3)))))
```

5a
```
(define (xor ci cj)
  (cond ((null? ci) cj)
        ((null? cj) ci)
        ((< (car ci) (car cj)) (cons (car ci) (xor (cdr ci) cj)))
        ((> (car ci) (car cj)) (cons (car cj) (xor ci (cdr cj))))
        ((= (car ci) (car cj)) (xor (cdr ci) (cdr cj)))))
```

5b
```
(define (enumere l)
  (define (aux x l1 l2 res)
    (cond ((null? l1) res)
          ((null? l2) (aux (car l1) (cdr l1) l res))
          (else (aux x l1 (cdr l2) (if (member (xor x (car l2)) res) res
                                       (cons (xor x (car l2)) res))))))
  (aux (car l) (cdr l) l ()))
```

EXAMEN INFORMATIQUE MIAS/MASS **FORMULAIRE DE REPONSE** **Juin 2003**

Numéro de place :

Chaque candidat écrit son nom dans le coin de la copie qu'il cachera par collage *après* l'émargement. Le numéro de place doit figurer sur chaque copie ou intercalaire.

1a **1b**

1c **1d**

2a (define (defaut t)
.
.
.

2b
.
.
.
.
.
.
.

3a **3b**

3c

4 (define (u n u0)
.
.
.
.

5a (define (xor ci cj)
.
.
.
.
.
.
.

5b (define (enumere l)
.
.
.
.
.
.
.