```
;;;;
;;;; Corrigé-Feuille7.stk       -- Corrigé de la feuille de Td n° 7
;;;;
;;;;           Author: Erick Gallesio [eg@essi.fr]
;;;;    Creation date: 19-May-2003 15:40 (eg)
;;;; Last file update: 19-May-2003 15:51 (eg)
;;;;

;==============================================================================
;                                    EXO 1
;==============================================================================

;;; C'est dans le cours !!!!



;==============================================================================
;                                    EXO 2
;==============================================================================

(define (creer-carte val)
  (lambda (msg . args)
    (case msg
      ((decrementer!)   (if (= val 0) (error "Plus de credit"))
                        (set! val (- val 1))
                        val)
      ((afficher-unite) val)
      (else             (error "message inconnu ~S\n" msg)))))


#|
(define C1 (creer-carte 50))
(C1 'decrementer!)
|#


(define (creer-carte-protegee val pass)
  (let ((mot-de-passe-ok #f)
        (carte           (creer-carte val)))
    (lambda (msg . args)
      (case msg
        ((password)     (if (equal? (car args) pass)
                            (set! mot-de-passe-ok #t)
                            (error "mot de passe errone")))
        ((decrementer!) (if mot-de-passe-ok
                            (apply carte msg args)
                            (error "Vous n'avez pas donne de mot de passe")))
        ((raccrocher)   (set! mot-de-passe-ok #f) 'Salut)
        (else           (apply carte msg args))))))

#|
(define C2 (creer-carte-protegee 50 1234))
(C2 'afficher-unite)
(C2 'password 12)
(C2 'password 1234)
(C2 'decrementer!)
(C2 'raccrocher)
|#

;==============================================================================
;                                    EXO 3
;==============================================================================
```



```
;;
;; Question 1 & 2
;;

(define-macro (define-record name fields)
  (define (make-reader slot)
    `((,(string->symbol (format #f "get-~A" slot))) ,slot))

  (define (make-writer slot)
   `((,(string->symbol (format #f "set-~A!" slot))) (set! ,slot (car args))))

 `(define (,name)
    (let ,(map (lambda (x) (list x #f)) fields)

      (lambda (msg . args)
        (case msg
          ((type)  ',name)
          ((print) (format #t "#<~A" ',name)
                   ,@(map (lambda (x)
                            `(format #t " ~A=~S" ',x ,x))
                          fields)
                   (format #t ">\n"))

          ;; Les lecteurs
          ,@(map make-reader fields)
          ;; Les écrivains
          ,@(map make-writer fields)
          ;; Le traitement d'erreur
          (else (error "Accesseur inconnu ~S" msg)))))))

(define (make-record prototype)         ; Version 1
  (prototype))

#|
Tests
=====

(define-record Point-2D
  (x y))

(pp (macro-expand '(define-record Point-2D  (x y))))

       ==>
(define (point-2d)
   (let ((x #f)
         (y #f))
     (lambda (msg . args)
       (case msg
         ((type)  'point-2d)
         ((print)  (format #t "#<~A" 'point-2d)
                   (format #t " ~A=~S" 'x x)
                   (format #t " ~A=~S" 'y y)
                   (format #t ">\n"))
         ((get-x)  x)
         ((get-y)  y)
         ((set-x!) (set! x (car args)))
         ((set-y!) (set! y (car args)))
         (else     (error "Accesseur inconnu ~S" msg))))))

(define p1 (make-record Point-2D))
(p1 'set-x! 10)
(p1 'print)
```

```
|#


;;
;; Question 3
;;

(define (make-record prototype . args)  ; Version 2
  (let ((res (prototype)))
    (do ((l args (cddr l)))
        ((null? l))
      (res (string->symbol (format #f "set-~A!" (car l))) (cadr l)))
    res))

#|
Tests
=====

(define p2 (make-record Point-2d 'y 20 'x 0))
(p2 'print)
( (make-record Point-2d 'x 1000) 'print)
|#
```