Lenguajes y Paradigmas de Programación
Curso 2003-2004
Examen de la Convocatoria de Diciembre

Normas importantes

- La puntuación total del examen es de 40 puntos que sumados a los 20 puntos de las prácticas dan el total de 60 puntos sobre los que se valora la nota de la asignatura.
- Para sumar los puntos de las prácticas **es necesario obtener un mínimo de 16 puntos en este examen**.
- La duración del examen es de 2 horas.

Pregunta 1 (8 puntos)
(6 puntos) Escribe un procedimiento llamado `power-close-to` que tome dos enteros positivos (`b` y `n`) como argumento y devuelva la potencia más pequeña de `b` que es mayor que `n`. Esto es, debería devolver el entero más pequeño *i* tal que *b^i > n*.

```
(power-close-to 2 100) -> 8
```

(2 puntos) ¿El procedimiento que has escrito es recursivo o iterativo?

Pregunta 2 (8 puntos)

Suponiendo que tree es un árbol cuyos datos son números, define el procedimiento `(suma tree)` que devuelva la suma de todos los datos del árbol tree. Debes usar los selectores definidos en clase:

```
(define (make-tree dato hijos)  (cons dato hijos))
(define (dato tree)  (car tree))
(define (hijos tree)  (cdr tree))
```

Por ejemplo, `(suma '(1 (2) (3 (4 (5)))))` debe devolver 15

Pregunta 3 (8 puntos)

oHe aquí un procedimiento para construir un objeto abstracto persona. El procedimiento está incompleto.

```
(define (make-persona nombre edad profesion dni)
   (let ((persona (list nombre edad profesion dni))
      (lambda (m)
         (cond ((eq? m ´nombre)   <a>)
               ((eq? m ´e-mail)   <b>)
               ((eq? m ´empresa)  <c>)
               ((eq? m ´tfno)     <d>)
               (else (error "petición desconocida" m))))))
```

(continúa detrás)

Un ejemplo de uso del procedimiento anterior serían las siguientes instrucciones:

```
(define persona-1
   (make-persona ´lucia ´lgl@yahoo.com ´ua ´966454321))
(define persona-2
   (make-persona ´david ´dvg@hotmail.com ´dccia ´953441234))
```

1. (4 puntos) Escribe qué sentencias Scheme deberían ir en los lugares marcados con <a>, <b>, <c>, <d>.
2. (4 puntos) ¿Cómo se preguntaría por el dni de una persona? Pon un ejemplo con el objeto persona-2.

Pregunta 4 (8 puntos)
Consideramos que las siguientes expresiones se evalúan en el orden que aparecen:

```
(define a (list (list ´q) ´r ´s))
(define b (list (list ´q) ´r ´s))
(define c a)
(define d (cons ´p a))
(define e (list ´p (list ´q) ´r ´s))
```

1. (4 puntos) Dibuja un diagrama box-and-pointer con las estructuras resultantes
2. (4 puntos) Indica cuál sería el resultado de las siguientes expresiones:

```
(eq? a c)                  (equal? a c)
(eq? a b)                  (equal? a b)
(eq? a (cdr d))            (equal? a (cdr d))
(eq? (car a) (cadr e))     (equal? (car a) (cadr e))
```

Pregunta 5 (8 puntos)
Supongamos los siguientes programas Scheme.

```
(define z 3)
(define h (let ((x 2))
             (lambda (y)
                (+ x y z))))
(h 4)
```

```
(define h
    (let ((x 2))
       (lambda (y)
         (let ((z 3))
            (+ x y z)))))
(h 4)
```

1. (4 puntos) Dibuja los entornos resultantes después de evaluarse el programa 1 y el 2 (ambos se evalúan por separado)
2. (1 punto) ¿Cuál es el valor que devuelve la última expresión en ambos casos?
3. (3 puntos) Dos entornos son equivalentes cuando todos los programas Scheme dan el mismo resultado en ambos. ¿Son equivalentes los entornos resultantes de evaluar el programa 1 y el 2? Si piensas que los entornos no son equivalentes, encuentra algún programa Scheme que se comporte de forma distinta (devuelva un valor distinto) en uno y otro entorno.