

Lenguajes y Paradigmas de Programación
Curso 2010-2011
Primer parcial

- La puntuación total del examen es de 20 puntos.
- Se debe contestar cada pregunta en las hojas que entregamos. Utiliza las últimas hojas para hacer pruebas. No olvides poner el nombre.
- La duración del examen es de 1 hora.

Apartado 1 (1 punto)

b) ¿La expresión resultante es una lista? Explica tu respuesta.

Apartado 2 (1 puntos)

Representa en forma de árbol las siguientes expresiones:

Expresión-s: (a ((b c (d)) e) f)

Árbol binario: (23 (12 () (20 () ())) (40 (30 () ())) ()))

Árbol genérico: (a (b (c) (d) (e (f))) (g))

Apartado 3 (2 puntos)

Explica en qué consisten los siguientes términos:

- Closure
- Función de orden superior
- Modelo de evaluación

Apartado 4 (2 puntos)

- a) Dibuja los ámbitos que representa la siguiente expresión
- b) ¿Qué valor devolverá Scheme?
- c) Escribe una expresión equivalente en Scheme que no utilice let

```
(define x 5)
(define y 2)
(let ((x (+ 1 2))
      (y (+ x 3)))
  (+ x y))
```

Ejercicio 2 (4 puntos)

Escribir la función `(rotar sentido lista)` que reciba un sentido de rotación (izquierda o derecha) y una lista y la rote hacia el sentido indicado. No puedes utilizar `length` ni `list-ref`. A continuación indica si tu función genera un proceso recursivo o iterativo.

Ejemplos:

```
(rotar '>' '(1 2 3 4)) -----> (4 1 2 3)
(rotar '<' '(1 2 3 4)) -----> (2 3 4 1)
```

Ejercicio 3 (5 puntos)

Define una función recursiva `(merge lista1 lista2)` que tome como argumentos dos listas ordenadas sin números repetidos y devuelva una lista (también sin números repetidos) en la que se han intercalado de forma ordenada los números de ambas listas.

Ejemplo:

```
(define lista1 '(1 2 8 12 23 56))  
(define lista2 '(3 8 10 22 23 48))  
(merge lista1 lista2) -> (1 2 3 8 10 12 22 23 48 56)
```

Ejercicio 4 (5 puntos)

- a) Define la barrera de abstracción del árbol genérico y su implementación.
- b) Utilizando esa barrera de abstracción, escribe el procedimiento `(filtra-tree f tree)` que reciba un árbol genérico y un predicado como argumentos y devuelva una lista con todos los nodos del árbol que cumplan el predicado.