

Nombre: _____ Grupo: _____

Lenguajes y Paradigmas de Programación

Curso 2011-2012

Primer parcial

Normas importantes

- La puntuación total del examen es de 10 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 2 horas.

Ejercicio 1 (1 punto)

a) (0,75 puntos) Indica las características principales de los siguientes paradigmas de programación y da al menos 1 ejemplo de lenguaje.

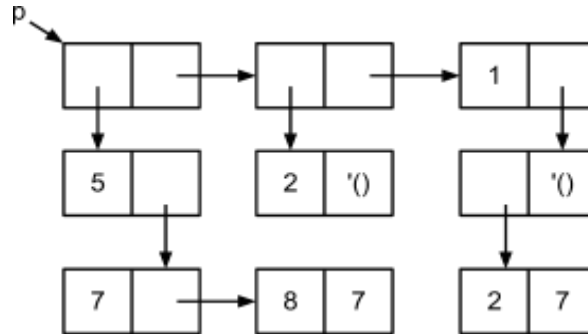
- Funcional
- Imperativo
- Orientado a Objetos
- Lógico

b) (0,25 puntos) ¿Qué es un lenguaje multiparadigma? Pon dos ejemplos.

Ejercicio 2 (0,75 puntos)

a) (0,5 puntos) Dado el siguiente diagrama Box & Pointer, escribe una expresión en Scheme que lo genere.

b) (0,25 puntos) Indica si es o no una lista, razonando tu respuesta.



Ejercicio 3 (1 punto)

Define la función (cumple-predicados lista-preds n) que recibe una lista de predicados y un número y devuelve una lista con los resultados de aplicar cada predicado al número. Ejemplo: (cumple-predicados (list even? mayor-5? menor-10?) 6) → (#t #t #t)

Ejercicio 4 (1 punto)

Dadas las siguientes definiciones:

```
(define (g f)
  (lambda (x) (f (f (f x)))))
(define (suma-1 x)
  (+ x 1))
```

- a) Escribe una expresión en Scheme que llame a g y que devuelva 5.
- b) Escribe lo que devolverá Scheme al evaluar la siguiente expresión:
`((g (g suma-1)) 0)`

Ejercicio 5 (2,5 puntos)

a) (1,5 puntos) Define las siguientes funciones recursivas (sin utilizar list-tail):

- (n-ultimos lista n): recibe una lista y un número. Devuelve una lista que contiene los n últimos elementos de la lista. Ejemplo:
(n-ultimos '(1 2 3 4 5 6) 3) → (4 5 6)
- (n-primeros lista n): recibe una lista y un número n. Devuelve una lista que contiene los n primeros elementos de la lista. Ejemplo:
(n-primeros '(1 2 3 4 5 6) 2) → (1 2)

b) (1 punto) Define la función (mover-n lista n) que devuelva una lista resultado de mover los n últimos elementos al principio. Puedes utilizar las funciones anteriores.

Ejemplo:

```
(mover-n '(a b c d e f g) 3) --> (e f g a b c d)
(mover-n '(a b c d e f g) 6) --> (b c d e f g a)
(mover-n '(a b c d e f g) 1) --> (g a b c d e f)
```

Ejercicio 6 (2,25 puntos)

a) (1 punto) Define la función (construye-poli lista-x lista-y) que reciba dos listas que representan cada una las coordenadas x e y de los vértices de un polígono, y devuelva una lista de parejas donde cada pareja representa la coordenada (x,y) de cada vértice.

Ejemplo:

```
(define poli (construye-poli '(3 7 5 9) '(-2 -6 4 6))) → ((3.-2)(7.-6)(5.4)
(9.6))
```

b) (1,25 puntos) Define la función (coords-min poli) que reciba un polígono en el formato anterior (lista de parejas) y devuelva una nueva pareja donde la coordenada x sea la mínima x de todas las del polígono y la y sea la mínima de todas las y's del polígono. Ejemplo:

```
(coords-min poli) → (3.-6)
```

Ejercicio 7 (1,5 puntos)

Dadas las siguientes expresiones:

```
(define x 10)
(define (g z)
  (let ((x 20)
        (y (+ x z)))
    (lambda (y) (+ x y z))))
(define a (g x))
(a x)
```

a) (0,25 puntos) Indica el resultado.

b) (1 punto) Dibuja y explica el diagrama de entornos asociado a su evaluación.

c) (0,25 puntos) ¿Se ha creado alguna *closure*? Explícalo.