

Lenguajes y Paradigmas de Programación

Curso 2007-2008

Examen de la Convocatoria de Septiembre

Normas importantes

- La puntuación total del examen es de 100 puntos sobre los que se valora la nota de la asignatura.
- Para sumar los puntos de las prácticas **es necesario obtener un mínimo de 40 puntos en este examen.**
- Se debe contestar cada pregunta **en una hoja distinta**. No olvides poner el nombre en todas las hojas.
- La duración del examen es de 3 horas.
- Las notas estarán disponibles en la web de la asignatura el próximo día 19 de Septiembre. La revisión se realizará la semana siguiente en una fecha que se anunciará en la web el mismo día en que estén disponibles las notas.

Pregunta 1 (16 puntos)

Escribe el procedimiento mutador (`split! lista`) que tome una lista como argumento y la divida en dos: la primera con los elementos de las posiciones impares de la lista y la segunda con los de las pares. La función no debe crear ninguna pareja nueva (con la excepción de la que se devuelve), sino que se debe modificar los punteros de la lista original.

```
(define l '(a b c d e f))
(define p (split! l))
(car p) → (a c e)
(cdr p) → (b d f)
l → (a c e)
```

Pregunta 2 (16 puntos)

Escribe el procedimiento mutador (`reemplazar! pred proc lista`) que reemplace cada elemento de la lista que cumpla el predicado `pred` por la aplicación del procedimiento `proc` correspondiente a ese elemento.

```
(define l '(1 2 3 4))
(reemplazar! even? square l)
l → '(1 4 3 16)
```

Pregunta 3 (18 puntos)

Supongamos el siguiente escenario:

“Necesitamos un sistema informático que gestione el calendario de un profesional que atiende a pacientes y les da citas.

- El sistema debe poder añadir y eliminar citas, así como detectar posibles solapes entre ellas.
- El sistema debe proporcionar los huecos libres de un determinado día.”

Realiza lo siguiente:

1. **(6 puntos)** Escribe la barrera de abstracción en Scheme de todos los tipos de datos que utilizarías para implementar este escenario.
2. **(6 puntos)** Implementa las funciones definidas en la barrera de abstracción (todas las que puedas).

3.(6 puntos) Escribe un conjunto de pruebas que validen todas las funcionalidades de la barrera de abstracción. Cada prueba debe consistir en un conjunto de instrucciones en Scheme que realizan una llamada a la barrera de abstracción utilizando datos concretos junto con una indicación de cuál es el resultado esperado.

Pregunta 4 (18 puntos)

Vamos a construir un procedimiento `maquina-del-tiempo` que recuerde los valores de los argumentos que se le han pasado. Cuando se llame al procedimiento `maquina-del-tiempo`, se guarda el valor de la llamada actual, y devuelve el valor que se pasó n veces antes, donde n es un número que se proporciona cuando la máquina del tiempo se construye. Si la máquina es nueva (se han hecho menos de n llamadas), se devuelve el valor inicial con el que la máquina se creó.

`(make-maquina-tiempo n valor_ini)` -> Crea una máquina del tiempo que devuelve valores de n llamadas anteriores, con valor inicial `valor_ini`.

Ejemplos:

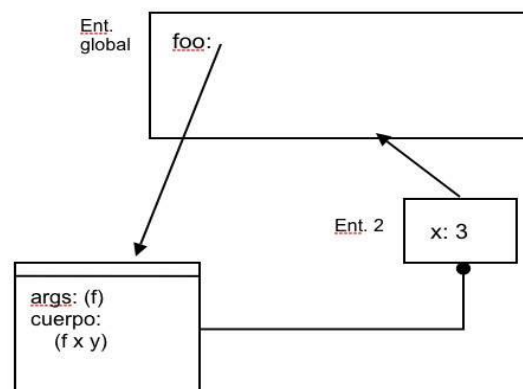
```
(define mt1 (make-maquina-tiempo 1 #f))
(mt1 5) → #f
(mt1 'a) → 5
(mt1 8) → a

(define mt2 (make-maquina-tiempo 2 'hola))
(mt2 5) → hola
(mt2 'a) → hola
(mt2 8) → 5

(define mt0 (make-maquina-tiempo 0 #f))
(mt0 5) → 5
(mt0 'a) → a
(mt0 8) → 8
```

Pregunta 5 (16 puntos)

Supongamos el siguiente diagrama de entornos:



1.(6 puntos) Escribe un conjunto de instrucciones que genere ese diagrama de entornos.

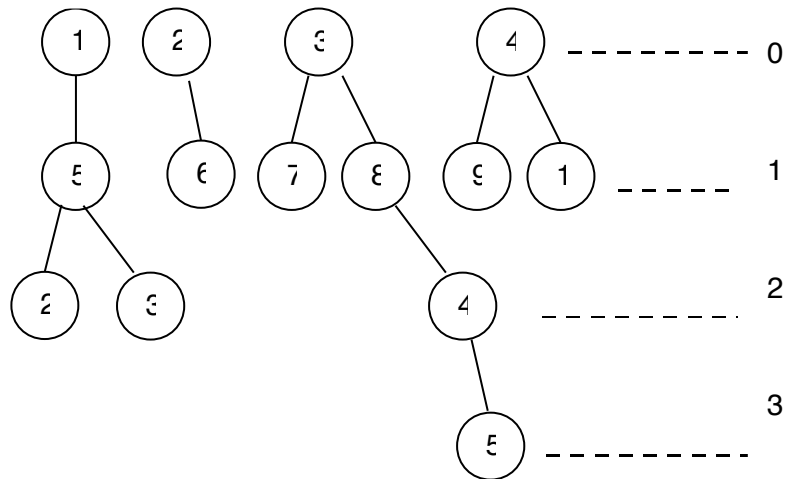
2.(5 puntos) Escribe un conjunto de instrucciones que, ejecutadas en el entorno global, tengan como resultado 15 y dibuja el diagrama de entornos resultantes.

3.(5 puntos) ¿Sería posible cambiar el valor de x en el entorno 2? Si crees que sí, indica cómo hacerlo escribiendo el código Scheme que haría el cambio.

Pregunta 6 (16 puntos)

Escribe el procedimiento `profundidad-forest` que devuelva la máxima profundidad de un bosque (lista de árboles).

bosque:



`(profundidad-forest bosque)`

3
