

Lenguajes y Paradigmas de Programación

Curso 2003-2004

Examen de la Convocatoria de Septiembre

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.**
- Se debe contestar cada pregunta **en un hoja distinta**. No olvides poner el nombre en todas las hojas.
- La duración del examen es de 2 horas.

Pregunta 1 (8 puntos)

Escribe una función (`palabras-repetidas frase1 frase2`) que tome dos frases y devuelva el número de palabras repetidas en ellas. En el caso de que una palabra esté repetida en la misma frase se debe contar en las múltiples ocasiones que aparece

Ejemplos:

```
(palabras-repetidas '(me llamo juan) '(juan es alto))  
1  
(palabras-repetidas '(hola que me me dices) '(me))  
2  
(palabras-repetidas '(hola hola hola adios) '(hola hola))  
6
```

Pregunta 2 (8 puntos)

Escribe una función denominada (`perfecto? numero`) que reciba como parámetro un número entero y devuelva true si y sólo si el número es perfecto. Un número es perfecto si es igual a la suma de sus divisores inferiores a él. Por ejemplo, el número 28 es perfecto porque $28 = 14 + 7 + 4 + 1$ y los divisores de 28 son (exceptuando el propio 28) 14, 7, 4 y 1.

Intenta hacer el ejercicio lo más modular posible, definiendo en su caso funciones auxiliares.

Pregunta 3 (8 puntos)

Suponemos árboles binarios definidos con la siguiente interfaz

```
(define (make-tree dato izq der)
  (list dato izq der))

(define (dato tree) (car tree))
(define (hijo-izq tree) (car (cdr tree)))
(define (hijo-der tree) (car (cdr (cdr tree))))
```

Define el procedimiento `(swap-hijos! tree)` que intercambie los hijos del árbol binario pasado como parámetro. El procedimiento debe modificar el árbol mutando sus hijos.

```
(define tree '(1 (2 (3) (4)) (4 (3) (4))))
(swap-hijos! tree)
tree
(1 (4 (3) (4)) (2 (3) (4)))
```

Pregunta 4 (8 puntos)

Dibuja **y explica** el modelo de entorno resultante de las siguientes instrucciones Scheme. **Numera los entornos en el orden en que se van creando.**

```
(define (mystery)
  (let ((baz 1000))
    (define (dispatch msg)
      (cond
        ((eq? msg 'one)
         (begin
          (set! baz (/ baz 2))
          baz))
        ((eq? msg 'two)
         (let ((m (dispatch 'one)))
          (set! baz (* m 4))
          baz))))
      dispatch))

(define x (mystery))

(x 'two)
```

(EL EXAMEN CONTINÚA DETRÁS)

Pregunta 5 (8 puntos)

a) Define un procedimiento (`scale-stream s f`) que recibe un stream de números `s` y devuelve otro stream con los números de `s` escalados por el factor `f`

```
(display-stream (scale-stream pairs 3))  
6  
12  
18  
24  
30  
...
```

b) Necesitamos generar un stream de potencias de `x`. Define el procedimiento (`powers x`) para que genere dicho stream. **Pista:** puedes usar el procedimiento anterior `scale-stream`.

```
(display-stream (powers 2))  
1  
2  
4  
8  
16  
32  
...
```