

Tarea No.4 de Fundamentos de programación

Ejercicios con listas

Ángela Villota Gómez



Septiembre 14 de 2007

Antes de empezar:

- Esta tarea debe ser resuelta en parejas.
- En caso de copia, los grupos involucrados tienen cero.
- Debe seguir en orden los pasos de la *Estrategia de diseño*.
- Cada uno de los ejercicios debe tener su contrato, propósito, ejemplos (mínimo 3), programa y pruebas.
- el lenguaje de DrScheme debe ser: **Estudiante principiante**.

Importante:

Toda función que use datos compuestos debe tener una definición de los datos

Las funciones que tengan condicionales deben llevar análisis de las condiciones

1. Ejercicios con listas

1. Desarrolle la función *convertir a dólar* que recibe una lista de montos en pesos y retorna una lista con la conversión a dólar de los montos de dicha lista. Las entradas son la lista y la tasa de cambio.
2. Desarrolle la función *quitar-carro* que saca de una lista los precios que son considerados caros. Para esto, la función tiene dos entradas, una lista de precios y un número que es el límite (del precio a pagar). La función retorna la lista de precios que son menores o iguales al límite.
3. Desarrolle la función *sustituir* que tiene 3 entradas: una lista de juguetes (símbolos) y dos símbolos (anterior y nuevo). La función debe retornar una lista en donde el símbolo anterior es cambiado por el nuevo.
Ejemplo:
(sustituir 'doll 'Barbie (list 'robot 'doll 'dress))
;; debe retornar (list 'robot 'Barbie 'dress)
4. Desarrolle la función *n-circulos* la cual tiene como entrada una estructura llamada **circulo** y un natural. La función pinta n círculos en el lienzo de dibujo. Para esto usted debe tener en cuenta las siguientes definiciones de datos y las funciones auxiliares:

- usar el paquete de enseñanza `draw.ss`.
- definir la estructura **circulo**.
- definir la función auxiliar **pintar-circulo** que toma como entrada un **circulo**, retorna true y pinta en el lienzo el círculo que pasó como entrada.
- definir la función **trasladar** que toma como entrada un **circulo** y retorna otro círculo con una traslación de 10 píxeles en el centro, el radio debe ser el mismo.

2. Funciones con listas

A continuación está el enunciado de las funciones con listas, estas deben servir para cualquier lista, esto quiere decir, para listas de distintos tipos de datos

1. Desarrolle la función *longitud* que toma una lista *l* como entrada y retorna la longitud de la lista *l*.
Ejemplo:
(longitud (list 1 2 3)) debe retornar 3
2. Desarrolle la función *elemento?* que tiene como entradas un elemento *e* y una lista *l*. La función retorna verdadero si el elemento está en la lista, en el caso contrario la función retorna falso.
Ejemplo:
(elemento? 1 (list 1 2)) debe retornar *true*
(elemento? 1 (list 2 3)) debe retornar *false*
3. Desarrolle la función *elemento-n*, que toma como entradas una lista y un natural *n*. La función retorna el *n*-ésimo elemento en la lista.
Ejemplo:
(elemento-n 2 (list 1 2 3)) debe retornar 2
4. Desarrolle la función *insertar-primero* que tiene como entradas: un elemento y una lista. La función retorna una lista en la que se ha insertado el elemento en la primera posición.
Ejemplo:
(insertar-primero 3 (list 1 2)) debe retornar (list 3 1 2)
5. Desarrolle la función *insertar-en-n* que tiene como entradas: un elemento, una lista y un natural *n*. La función

retorna una lista en la que se ha insertado el elemento en la n-esima posición.

Ejemplo:

(insertar-en-n 5 (list 1 2 3 4) 2) debe retornar (list 1 5 2 3 4)

6. Desarrolle la función *insertar-ultimo* que tiene como entradas: un elemento y una lista. La función retorna una lista en la que se ha insertado el elemento en la última posición.

Ejemplo:

(insertar-ultimo 3 (list 2 1)) debe retornar (list 2 1 3)

7. Desarrolle la función *inversa* que tiene como entrada una lista y retorna la inversa de la lista de entrada.

Ejemplo:

(inversa (list 1 2 3 4)) debe retornar: (list 4 3 2 1)

8. Desarrolle la función *concatenar* que toma como entrada dos listas y retorna una lista con la concatenación de dichas listas.

Ejemplo:

(concatenar (list 1 2) (list 3 4)) debe retornar (list 1 2 3 4)