

Instituto Tecnológico de Durango

Departamento de Sistemas y Computación

Programación Lógica y Funcional

Trabajo Final

Resuelve los siguientes ejercicios.

Ejercicio 1. Sobre parejas. Predice lo que hace Scheme cuando escribas las siguientes expresiones. Están ordenadas por dificultad de arriba abajo y de izquierda a derecha. Después, pruébalas y comprueba si tu predicción era correcta. Si no lo era, intenta comprender por qué.

Instrucción	Instrucción
(cons 1 2)	(car (car (cons (cons 1 2) 3)))
(car (cons 1 2))	(car (cons (cons 3 4) 2))
(cdr (cons 1 2))	(cdr (cons (cons 3 4) 2))
(cons (* 2 3) (/ 4 2))	(cdr (cons 1 (cons 2 3)))
(cons (+ 2 1) (if (> 2 3) "2" "3"))	(cdr (car (cons (cons 1 2) 3)))

Ejercicio 2. Sobre listas. Predice lo que hace Scheme cuando escribas las siguientes expresiones. Después, pruébalas y comprueba si tu predicción era correcta. Si no lo era, intenta comprender por qué.

Instrucción	Instrucción
(list 1 2 3 4)	(cons 3 '(1 2 3))
(rest (list 1 2 3 4))	(rest (cons #t (cons "Hola" (list 1))))
(first '(1 2 3 4))	(first (list (list 1 2) 1 2 3 4))
(first (list #t 1 "Hola"))	(first (rest '((1 2) 1 2)))
(first (rest (list 1 2 3 4)))	(cons '(1 2 3) '(4 5 6))
(rest (rest '(1 2 3 4)))	(first (rest (list 1 2 3 4)))
(first (rest (rest (list 1 2 3 4))))	(rest (rest (list 1 2 3 4)))
(list (* 2 2) (+ 1 2) (/ 4 2))	(first (rest (rest (rest '(1 2 3 4)))))

Ejercicio 3. Sobre listas. Intenta hacer los siguientes apartados sin utilizar el intérprete de Scheme. Después comprueba si has acertado.

a) Dada la siguiente lista, indica la expresión correcta para que Scheme devuelva 3:

(list 1 2 3 4 5)

b) Dada la siguiente lista, indica la expresión correcta para que Scheme devuelva (5).

(list 1 2 3 4 5)

c) Dada la siguiente lista, indica la expresión correcta para que Scheme devuelva 5.

(list 1 2 3 4 5)

d) Dada la siguiente expresión, ¿qué devuelve Scheme?

```
(first (rest (rest (list 1 (list 2 3) (list 4 5) 6))))
```

e) Dada la siguiente expresión, ¿qué devuelve Scheme?

```
(rest (rest '(1 (2 3) 4 5)))
```

Ejercicio 4. Predice lo que devolverá Scheme cuando escribas las siguientes expresiones. Están ordenadas por dificultad de arriba abajo y de izquierda a derecha. Después, pruébalas y comprueba si tu predicción era correcta. Si no lo era, intenta comprender por qué.

Instrucción	Instrucción
(equal? "hola" "hola")	(+ (char->integer(integer->char 1200)) (char->integer #\A))
(string-ref "pepe" 1)	(string-length (make-string 7 #\E))
(substring "buenos dias" 1 4)	(define a 3) (define b (+ a 1))
(= "hola" "hola")	(+ a b (* a b))
(string-ref (substring "buenos dias" 2 5) 1)	(= a b)
(define pi 3.14159)	(if (and (> a b) (< b (* a b))) b a)
pi	(cond ((= a 4) 6) ((= b 4) (+ 6 7 a)) (else 25))
"pi"	(+ 2 (if (> b a) b a))
(+ pi (+ pi pi))	(* (cond ((> a b) a) (< a b) b) (else -1)) (+ a 1))
(+ (* pi pi) (- 2 pi pi pi pi))	((if (< a b) + -) a b)

Ejercicio 5. Define una función que calcule la distancia entre dos puntos, definidos por parejas de números enteros. Añade los siguientes tests para comprobar que funciona correctamente:

Entrada	Salida
p1:(0 0) p2:(0 10)	10
p1:(0 0) p2:(10 0)	10
p1:(0 0) p2:(10 10)	14.142135623730951

Ejercicio 6. Resuelve lo que se te solicite.

a) Implementa la función (binario-a-decimal b3 b2 b1 b0) que reciba 4 bits que representan un número en binario y devuelva el número decimal equivalente.

(binario-a-decimal 1 1 1 1) \Rightarrow 15

(binario-a-decimal 0 1 1 0) \Rightarrow 6

(binario-a-decimal 0 0 1 0) \Rightarrow 2

Nota: recuerda que para realizar esta conversión, se utiliza la siguiente fórmula:

$$n = b_3 * 2^3 + b_2 * 2^2 + b_1 * 2^1 + b_0 * 2^0$$

Para la implementación de la expresión debes utilizar la función `expt`.

b) Implementa la función (binario-a-hexadecimal b3 b2 b1 b0) que reciba 4 bits de un número representado en binario y devuelva el carácter correspondiente a su representación en hexadecimal.

(binario-a-hexadecimal 1 1 1 1) \Rightarrow #\F

(binario-a-hexadecimal 0 1 1 0) \Rightarrow #\6

(binario-a-hexadecimal 1 0 1 0) \Rightarrow #\A

Nota: para realizar esta conversión, como paso intermedio debes pasar primero el número binario a su representación decimal (utilizando la función definida en el apartado anterior) y después a su correspondiente hexadecimal.

Recuerda que la representación hexadecimal de los números decimales del 0 al 9 es el carácter correspondiente a ese número, y que el número decimal 10 se representa con el carácter A, el 11 con el B, y así sucesivamente hasta el 15 que es el F en hexadecimal.

Para la implementación de esta función auxiliar que pasa de decimal a hexadecimal debes usar las funciones `integer->char` y `char->integer`. En la función `char->integer` los caracteres consecutivos están asociados con números consecutivos. Por ejemplo, el entero correspondiente al carácter #\A es uno menos que el correspondiente al carácter #\B. Los caracteres de números y los de letras no son consecutivos.

Ejercicio 7. Resuelve lo que se te solicite.

Implementa la función (menor-de-tres n1 n2 n3) que reciba tres números como argumento y devuelva el menor de los tres, intentando que el número de condiciones sea mínima.

No debes utilizar la función `min`.

Implementa dos versiones de la función:

- versión 1: usando la forma especial `if`
- versión 2 (llámala `menor-de-tres-v2`): sin usar la forma especial `if`, sino definiendo una función auxiliar (`menor x y`) que devuelva el menor de dos números (en esta sí que deberías usar `if`) y construyendo la función `menor-de-tres-v2` como una composición de llamadas a esta función auxiliar.

`(menor-de-tres 2 8 1)` $\Rightarrow 1$

`(menor-de-tres-v2 3 0 3)` $\Rightarrow 0$

Ejercicio 8. Resuelve lo que se te solicite.

a) Supongamos las definiciones

```
(define (f x y)
  (cons x y))

(define (g x)
  (cons 2 x))
```

Realiza la evaluación paso a paso de la siguiente expresión

`(f (g (+ 2 1)) (+ 1 1))`

mediante el modelo de sustitución, utilizando tanto el orden aplicativo y como el orden normal.

b) Supongamos las definiciones

```
(define (f x)
  (/ x 0))

(define (g x y)
  (if (= x 0)
      0
      y))
```

Igual que en el apartado anterior, realiza la evaluación paso a paso de la siguiente expresión

`(g 0 (f 10))`

mediante el modelo de sustitución, utilizando tanto el orden aplicativo y como el orden normal.

Ejercicio 9. Resuelve lo que se te solicite.

Implementa la función `(cadenas-mayores lista1 lista2)` que recibe 2 listas con 3 cadenas y devuelve otra lista con las 3 cadenas de mayor longitud, comparando las cadenas de cada posición de la lista. En el caso en que las cadenas tengan la misma longitud, se devuelve la cadena de la primera lista.

`(cadenas-mayores '("hola" "que" "tal") ('("meme" "y" "adios")))` ; \Rightarrow `("hola" "que" "adios")`

`(cadenas-mayores '("esto" "es" "lpp") ('("hoy" "hay" "clase")))` ; \Rightarrow `("esto" "hay" "clase")`

Ejercicio 10. Resuelve lo que se te solicite.

Supongamos que queremos programar un juego de cartas. Lo primero que debemos hacer es definir una forma de representar las cartas y funciones que trabajen con esa representación. En este ejercicio vamos a implementar esas funciones.

Representaremos una carta por un símbolo con dos letras: la primera indicará su número o figura y la segunda el palo de la carta.

Por ejemplo:

(define tres-de-oros '3O)

(define as-de-copas 'AC)

(define caballo-de-espadas 'CE)

Debemos definir la función carta que devuelve una pareja con el valor correspondiente a su orden en la baraja española (un número) y el palo de la carta (un símbolo).

(carta tres-de-oros) ⇒ (3 . Oros)

(carta as-de-copas) ⇒ (1 . Copas)

(carta 'RB) ⇒ (12 . Bastos)

Los valores de las cartas de la baraja española son:

A (As) ⇒ 1

S (Sota) ⇒ 10

C (Caballo) ⇒ 11

R (Rey) ⇒ 12

Para realizar el ejercicio debes definir en primer lugar las funciones (obten-palo char) y (obten-valor char) que devuelven el palo y el valor, dado un carácter. Y debes implementar la función carta usando estas dos funciones.

(obten-palo #\O) ⇒ Oros

(obten-palo #\E) ⇒ Espadas

(obten-valor #\3) ⇒ 3

(obten-valor #\S) ⇒ 10

Pista: Puedes utilizar las funciones (symbol->string simbolo) que convierte un símbolo en una cadena y (string-ref cadena pos) que devuelve el carácter de una cadena situado en una determinada posición.