Lenguajes de Programación
Practica 1

Karla Ramírez Pulido José Eliseo Ortíz Montaño

Semestre 2024-1
Facultad de Ciencias UNAM

Fecha de inicio: 17 de agosto 2023
Fecha de entrega: 25 de agosto 2023

Instrucciones

La realización y entrega de la práctica deberá realizarse en equipos de a lo más 3 personas*. La entrega debe respetar el orden de las funciones que es especificado en este documento. El uso de funciones auxiliares está totalmente permitido, sin embargo, deben ser declaradas justo después de la función principal que hace uno de ellas.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Ejercicio 1: [Descripción del ejercicio]
(define (my-fun a b c d) ... )
;; [Documentación de la función auxiliar - incluir una breve descripción de
;; la motivación de la función]
(define (an-aux-fun lst1 lst2) ... )
...
...
...
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Las funciones deben incluir comentarios, tanto de documentación como del proceso de desarrollo. Estos deben ser claros, concisos y descriptivos.

Queda estrictamente prohibido utilizar funciones primitivas del lenguaje que resuelvan directamente los ejercicios.

Se deberá subir la versión final de su práctica al apartado del classroom correspondiente antes de la fecha límite. Esto solo debe realizarlo un integrante del equipo; el resto deberá marcar como entregada la actividad y en un comentario privado especificar quienes son los miembros del equipo.
El archivo debe ser llamado `practica1.rkt` y en forma de documentación, deberán incluir los datos de los miembros del equipo.

*Cualquier situación con respecto a este punto será tratada de acuerdo a las particularidades del caso. Para esto, acercarse al ayudante del rubro del laboratorio a la brevedad.

Ejercicios

1. Definir la función `area-total` que dada la generatriz y el diámetro de la base de un cono circular recto, calcular el área total del mismo.

$$At = \pi r g + \pi r^2$$

```
;; area-total :: number number ->number
(define (area-total g d) ... )
>(area-total 5 7)
93.46238144429634
>(area-total 3 8)
87.96459430051421
```

2. Definir el predicado `decremental?` que dados cuatro números indica si se encuentran ordenados de forma decremental.

```
;; decremental? :: number number number number ->boolean
(define (decremental? a b c d) ...)
 >(decremental? 1 2 3 4)
#f
>(decremental? 10 7 8 11)
#f
>(decremental? 10 9 8 7)
#t
```

3. Definir la función `multiplica` que, dada una lista, multiplica todos los elementos contenidos en la misma. **Hint:** es muy similar a la función que suma todos los elementos de una lista que trabajamos en clase.

4. Definir la función `area-heron` que calcula el área de un triángulo a partir de la medida de cada uno de sus lados. La fórmula de Herón es la siguiente:

A $= \sqrt{S(S-a)(S-b)(S-c)}$

donde S es el semiperímetro definido por:

S $= \frac{a+b+c}{2}$

```
>(area-heron 3 25 26)
36
>(area-heron 3 4 5)
6
>(area-heron 10.09 3.83 10.46)
19.24
```

5. Definir el predicado pares? que dada una lista, indica si todos los elementos contenidos en ésta son pares.

```
;; pares? :: (listof number) ->number
(define (pares? lst) ... )

>(pares? '(2 6 8 10 4 6))
#t
>(pares? '(45 2 4 8))
#f
```

6. Define la función filtra-lista que filtre el contenido de una lista dado un predicado.

```
;; filtra-lista :: (listof any) ->(listof any)
(define (filtra-lista p ls) ...)

>(filtra-lista number? '(2 "33"8 "10"4 6))
'(2 8 4 6)
>(filtra-lista string? '("2"2 4 "hola"))
'("2hola")
```