

INF-253 Lenguajes de Programación

Mini-tarea 4

Profesor: Roberto Nicolás Díaz Urrea
Ayudante Cátedras: Sebastián Godínez
Ayudante Tareas: Monserrat Figueroa - Sebastián Campos

1. Descripción

Esta mini-tarea consiste en resolver tres problemas donde cada uno corresponde a la implementación de una función en el lenguaje Scheme. Para cada problema se entregará un archivo de plantilla con extensión `.scm` para la resolución y entrega de la tarea. Este archivo deberá ser completado de forma de resolver cada problema.

El archivo de plantilla incluirá algunos casos de prueba con los que se podrá verificar la solución. Los casos de prueba incluidos no son exhaustivos y la solución debe poder aplicarse a otros casos.

Aunque en la descripción de cada función se establecen restricciones para los parámetros no es necesario verificar que estos las cumplan en su solución.

Se pueden usar funciones auxiliares siempre que se incluyan en los archivos donde son utilizadas.

2. Problemas a Resolver

A continuación se presentan los distintos problemas a resolver y sus puntajes asociados.

1. Par-Impar

- **Sinopsis:** (`par-impar ls`)
- **Puntaje:** 30 pts
- **Archivo:** `par-impar.scm`
- **Descripción:** Utilizando **recursión de cola**, la función `par-impar` recibe una lista `ls` con números enteros y retorna una lista con dos sublistas: una con los números pares y otra con los números impares.
- **Ejemplo:**

```
>(par-impar '(1 2 3 4 5 6 7 8 9))  
((2 4 6 8) (1 3 5 7 9))
```

2. Minimizar

- **Sinopsis:** (`minimizar proc ls`)
- **Puntaje:** 40 pts
- **Archivo:** `minimizar.scm`

- **Descripción:** La función `minimizar` recibe un procedimiento `proc`, el que recibe dos números reales y retorna un número real, y una lista `ls` con números reales distintos de largo mínimo 2. La función retorna una lista con los dos elementos de la lista `ls` que al ser pasados como parámetros a `proc` minimicen su valor, respecto de los otros pares de elementos de la lista. El orden de los parámetros en el retorno debe corresponder al orden en que los recibe `proc`. Si hay más de un par candidato se puede retornar cualquiera.
- **Ejemplo:**

```
>(minimizar (lambda (x y) (+ x y)) '(8 1 6 7 2 3))
(1 2)
>(minimizar (lambda (x y) (- x y)) '(8 9 6 1 7 5))
(1 9)
```

3. Abundante

- **Sinopsis:** (`abundante x`)
- **Puntaje:** 30 pts
- **Archivo:** `abundante.scm`
- **Descripción:** La función `abundante` recibe un entero `x`, tal que $x \geq 2$, y retorna un valor booleano. Retornará verdadero si es que `x` es un número abundante.
Un número es abundante si la suma de sus divisores es mayor al doble del número. Por ejemplo, 8 no es un número abundante pues la suma de sus divisores es $1 + 2 + 4 + 8 = 15 < 16$, pero 12 si es un número abundante pues $1 + 2 + 3 + 4 + 6 + 12 = 28 > 24$.
- **Ejemplo:**

```
>(abundante 8)
#f
>(abundante 12)
#t
```

3. Entrega

- Se debe trabajar de manera individual.
- La entrega debe realizarse en un solo archivo `.tar.gz` y debe llevar el nombre: **mini-tarea-4-ROL.tar.gz**
- El archivo `.tar.gz` debe contener un archivo `README.txt` que incluya el nombre y rol del estudiante, junto a los tres archivos de plantilla con los problemas resueltos.
- La entrega será vía Moodle y el plazo máximo de entrega es hasta el **06 de enero a las 15:40 hrs.**
- Por cada 10 minutos de atraso en la entrega habrá un descuento de 20 puntos.
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.