

Instrucciones:

- La fecha de entrega es el día: Lunes 23 de agosto a las 8am.
- Cada día de atraso implicará una pérdida de 20 puntos.
- Las tareas son estrictamente de carácter individual, tareas iguales se les asignará cero puntos.
- Deben incluir los casos de prueba que se solicitan en los ejercicios.

## TAREA NÚMERO 1

1. [10 puntos] Escriba una función recursiva en Scheme para la función de Ackerman:

$$A(N, M) = \begin{cases} 0 & \text{si } M = 0 \\ 2M & \text{si } N = 0 \\ 2 & \text{si } M = 1 \\ A(N - 1, A(N, M - 1)) & \text{en otro caso} \end{cases}$$

Luego calcule:  $A(1, 10)$ ,  $A(2, 4)$ ,  $A(3, 3)$

2. [15 puntos] Programe en Scheme con recursión y recursión lineal una función, denominada `collatz(x)`, que retorne la **cantidad** de repeticiones necesarias para que el valor llegue a 1 según la **Conjetura de Collatz**. Esta conjetura dice que, sea  $n$  un número inicial, tras un número finito de repeticiones de la operación se llega a 1. La operación se puede resumir en lo siguiente:

- Toma cualquier número entero positivo  $n$ .
- Si  $n$  es par, divida  $n$  entre 2.
- Si  $n$  es impar, multiplique  $n$  por 3 y sume 1, es decir,  $3n + 1$ .
- Repita el proceso indefinidamente hasta que el valor sea 1.

Luego calcule: `collatz(16)`, `collatz(x12)` y `collatz(31)`.

3. [15 puntos] Programe en Scheme con recursión y recursión lineal la siguiente función:

$$u_n = \begin{cases} 2u_{n-1} - 7u_{n-2} - 6u_{n-3} & \text{si } n \geq 3 \\ 1 & \text{si } n = 0 \\ 0 & \text{si } n = 1 \\ -1 & \text{si } n = 2 \end{cases}$$

Luego calcule:  $u(4)$ ,  $u(12)$ ,  $u(23)$ .

4. [30 puntos] Usando:

$$\prod_{i=a}^n f(i) = f(a)f(a+1) * * * f(n)$$

Escriba en Scheme funciones de las tres formas (Normal, Término, Lambda) para:

$$\prod_{i=a}^n (i^3 + 7i - 11)^2$$

Luego calcule:  $p(1, 2)$ ,  $p(1, 5)$ ,  $p(4, 7)$ .

5. [30 puntos] Un número perfecto es un número entero positivo cuya suma de todos sus factores enteros positivos es igual a el mismo. Por ejemplo, el 6 es un número perfecto ya que sus factores son (1, 2, 3) y la suma de estos es 6. Se clasifica como abundante si la suma de los factores es mayor al número y defectivo si la suma es menor al número. Con la ayuda del comando `let` escriba un programa en Scheme que contenga lo siguiente:

- a) **es-factor?**: Determina si un valor es factor o no de otro. Puede investigar y utilizar sobre la función *remainder*.
- b) **suma-factores**: Obtiene la suma de los factores de un número. Debe programarla con recursión o recursión lineal.
- c) **clasificacion**: Retorne uno de los siguientes tres **símbolos**: **'perfecto**, **'defectivo** o **'abundante**.
- d) **resultado**: Debe de leer de teclado el valor a evaluar y desplegar el tipo de número que es el valor ingresado.