



Taller de Álgebra I - Recuperatorio

PRIMER CUATRIMESTRE 2017

12 de julio de 2017

Aclaraciones

- El parcial se aprueba con tres ejercicios bien resueltos.
- Programe todas las funciones en lenguaje Haskell. El código debe ser autocontenido. Si utiliza funciones que no existen en Haskell, debe programarlas.
- Incluya la signatura de todas las funciones que escriba.
- No está permitido: alterar los tipos de datos presentados en el enunciado – utilizar técnicas no vistas en clase para resolver los ejercicios – utilizar listas.

Ejercicio 1

Determinar la signatura e implementar las siguientes funciones $f : \mathbb{Z} \rightarrow \mathbb{Z}$ y $g : \mathbb{Z} \rightarrow \mathbb{Z} \times \mathbb{Z}$:

$$f(n) = \begin{cases} n^2 & \text{si } n \text{ es múltiplo de } 5 \\ n - 1 & \text{en caso contrario} \end{cases},$$

$$g(n) = (3n, n^3).$$

Determinar la signatura e implementar la función $h(n) = g(f(n))$.

Ejercicio 2

Implementar una función `soloImpares` que dado $n \in \mathbb{N}_{>0}$ determine si n esta formado únicamente por dígitos impares.

Por ejemplo:

`soloImpares 33571` \rightsquigarrow `True`

`soloImpares 23579` \rightsquigarrow `False`

`soloImpares 101` \rightsquigarrow `False`

Ejercicio 3

Implementar la función `s :: Integer -> Integer -> Integer` definida para $n, m \in \mathbb{N}_{>0}$ por la siguiente fórmula:

$$s(n, m) = \sum_{i=1}^n i^3 + \prod_{i=1}^m (i^2 + 2i)$$

Por ejemplo:

`s 1 1` \rightsquigarrow `4`

`s 3 2` \rightsquigarrow `60`

Ejercicio 4

Implementar una función `piAprox :: Integer -> Float` que dado un $n \in \mathbb{N}_{>0}$ aproxime el valor del número π a partir de la suma de los n primeros términos de la serie de Gregory-Leibniz:

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

Es decir, para $n \in \mathbb{N}_{>0}$ la aproximación es $\pi \cong \frac{4}{1} - \frac{4}{3} + \dots \pm \frac{4}{2n-1}$ (cuando n es impar el último término está sumando y cuando n es par el último término está restando).

Por ejemplo:

`piAprox 1` \rightsquigarrow `4.0`

`piAprox 2` \rightsquigarrow `2.6666665`

`piAprox 7` \rightsquigarrow `3.283738484`

Ejercicio 5

Implementar la función `sumaRacionales :: Integer -> Integer -> Float` que dados $a, b \in \mathbb{N}_{>0}$ calcule la suma de todos los números racionales de la forma $\frac{p}{q}$ con $1 \leq p \leq a$ y $1 \leq q \leq b$. Por ejemplo:

`sumaRacionales 2 3` \rightsquigarrow $\frac{1}{1} + \frac{2}{1} + \frac{1}{2} + \frac{2}{2} + \frac{1}{3} + \frac{2}{3} = 5.5$