

Nombre y apellido:
Carrera:

L.U. o D.N.I.:
Número de orden:

Cant. de hojas:

Departamento de Computación – FCEyN – UBA

Taller de Álgebra I - Parcial

PRIMER CUATRIMESTRE 2017 – TURNO NOCHE

31 de mayo 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

Dar el tipo e implementar la función `elDelMedio` que dados $a, b, c \in \mathbb{Z}$ devuelva el número del medio al ordenarlos de menor a mayor. *Por ejemplo:*

`elDelMedio 2 1 10 ~ 2` `elDelMedio 3 3 (-10) ~ 3`

Ejercicio 2

Implementar una función `sonCongruentesModulo :: Integer -> Integer -> Integer -> Bool` que dados $a, b \in \mathbb{N}_{\geq 0}$, $m \in \mathbb{N}_{> 0}$ determine si $a \equiv b \pmod{m}$. No se permite usar la funciones `mod` y `div` de Haskell. *Por ejemplo:*

`sonCongruentesModulo 8 5 3 ~ True` (ya que $8 \equiv 5 \pmod{3}$)

`sonCongruentesModulo 8 6 3 ~ False`

Ejercicio 3

Para $n \in \mathbb{N}_{> 0}$ se define la sucesión:

$$a_n = 2 + \frac{1}{2 + \frac{1}{\ddots \frac{1}{2 + \frac{1}{2 + \frac{1}{2}}}}} \quad (\text{aparece } n \text{ veces el } 2).$$

Implementar una función `raizDe2Aprox :: Integer -> Float` que dado $n \in \mathbb{N}_{> 0}$ devuelva la aproximación de $\sqrt{2}$ definida por $\sqrt{2} \approx a_n - 1$. *Por ejemplo:*

`raizDe2Aprox 1 ~ 1` `raizDe2Aprox 2 ~ 1,5` `raizDe2Aprox 3 ~ 1,4`

Ejercicio 4

Implementar una función `esSumaDePerfectos :: Integer -> Bool` que dado $n \in \mathbb{N}_{\geq 0}$ determine si n es suma de dos cuadrados perfectos. Un número entero es un cuadrado perfecto si es el resultado de elevar un número entero al cuadrado. **Ayuda:** pueden utilizar la función `sqrt_n :: Integer -> Integer` que determina la parte entera de la raíz cuadrada de un número.

Por ejemplo:

`esSumaDePerfectos 13 ~ True` (ya que $13 = 9 + 4$) `esSumaDePerfectos 15 ~ False`

`esSumaDePerfectos 25 ~ True` (ya que $25 = 25 + 0$)

Ejercicio 5

Implementar una función `esCapicua :: Integer -> Bool` que dado $n \in \mathbb{N}_{\geq 0}$ determine si n es un número capicúa.

Por ejemplo: `esCapicua 1212 ~ False` `esCapicua 3773 ~ True` `esCapicua 3 ~ True`