
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

SEGUNDO CUATRIMESTRE 2017 – TURNO VIERNES

20 de octubre 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.

Ejercicio 1

Implementar la función `menorLex :: (Float, Float, Float) -> (Float, Float, Float) -> Bool` que dados dos vectores $x, y \in \mathbb{R}^3$ decida si x es menor a y en el sentido lexicográfico.

Por ejemplo:

```
menorLex (3,-1,2) (5,10,0) ~> True      (pues 3 < 5)
menorLex (4,-1,7) (4,21,5) ~> True      (pues sus primeras coordenadas coinciden y -1 < 21)
menorLex (2,1,31) (2,1,-5) ~> False
```

Ejercicio 2

Implementar una función `sumaIncompleta :: Integer -> Integer` que para cada $n \geq 1$ calcule $\sum_{k \in K_n} k^4$, donde K_n es el conjunto definido por

$$K_n = \{k \in \mathbb{Z} : 1 \leq k \leq 2n + 1, k \neq n\}.$$

Ejercicio 3

Implementar una función `coprimos :: Integer -> Integer -> Bool` que dados dos números enteros x e y nos contesta si tienen algún divisor en común mayor estricto que 1.

Por ejemplo:

```
coprimos 2 24 ~> False      (ambos son divisibles por 2)
coprimos 12 35 ~> True
coprimos 100 101 ~> True
```

Ejercicio 4

Implementar una función `cantidadMultiplosN :: [Integer] -> Integer -> Integer` que, dada una lista de enteros xs y un entero n , dice cuántos números de xs son múltiplos de n .

Por ejemplo:

```
cantidadMultiplosN [1 2 3 4 5 6 7 8] 2 ~> 4
cantidadMultiplosN [1 2 3 4 5 6 7 8] 3 ~> 2
cantidadMultiplosN [1 2 1 11 23 45 33 10] 5 ~> 2
```

Ejercicio 5

Implementar una función `sonTodasFibo :: [[Integer]] -> Bool` que dada una lista de listas de enteros indique si todas las listas están construidas por términos sucesivos de Fibonacci.

Por ejemplo:

```
sonTodasFibo [[1, 2, 3, 5, 8],[21, 34, 55, 89, 144, 233],[8, 13, 21, 34]] ~> True
sonTodasFibo [[1, 2, 3, 5, 8],[21, 34, 233, 89, 144, 55],[8, 13, 21, 34]] ~> True
```