Taller Práctico Haskell

Módulo

Debe encontrar el orden del entero 8 módulo 13, que se define como:

$$ord(8) = min(t : 8t \equiv 1 \mod 13)$$

es decir, debe encontrar el entero t más pequeño que satisfaga la congruencia:

$$8t \equiv 1 \mod 13$$

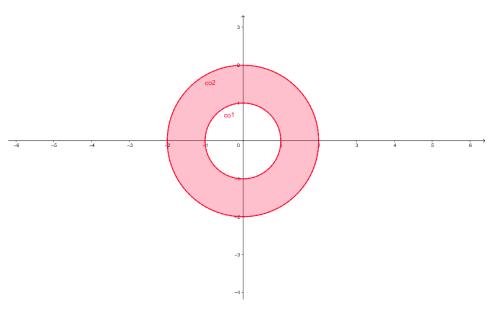
teniendo en cuenta que $t \in \{1, 2, 3, \dots, 13\}$

Funciones y Listas

• Definir la función areaDeCoronaCircular tal que areaDeCoronaCircular r1 r2 es el área de unacorona circular de radio interior r1 y radio exterior r2.

Pista:

areaDeCoronaCircular 1 2 == 9.42477796076938



• Implemente una función la cual tiene como argumento de entrada un valor numérico *n*, dicha función calculará la siguiente recurrencia:

$$0 * 1 + 2 * 3 + 4 * 5 + 6 * 7 + \cdots + (n - 1) * n$$

Su función debe retornar una lista que contenga la solución de la recurrencia desde n hasta 0, ordenado desde el último valor hasta el primero.

Pista:

$$n = 5 \rightarrow [50, 10, 6, 2, 0, 0]$$

 $n = 10 \rightarrow [3610, 3600, 400, 392, 56, 50, 10, 6, 2, 0, 0]$

• Definir la función palíndroma que dada una lista se verifica si sus elementos cumplen con la definición de un palíndromo.

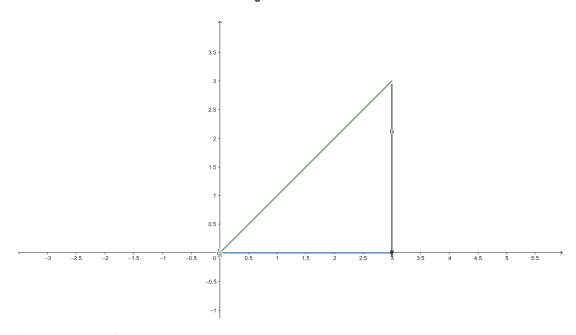
Pista:

Es lo mismo leer la lista de izquierda a derecha que de derecha a izquierda.

 Las longitudes de los lados de un triángulo no pueden ser cualesquiera. Para que pueda construirse el triángulo, tiene que cumplirse la propiedad triangular; es decir, longitud de cada lado tiene que ser menor que la suma de los otros dos lados. Definir la función triangular tal que (triangular a b c) se verifica si a, b y c cumplen la propiedad triangular.

Pista:

Triangular 3 4 5== True



Recursión y Comprensión

• Definir por recursión la función potencia :: Integer -> Integer tal que potencia x n es x elevado al número entero n.

Pista:

potencia 2 3 == 8

 Definir, por comprensión, la función sumaDigitosCadena :: String -> Int tal que sumaDigitosCadena x es la suma de los dígitos de la cadena x.

Pista:

sumaDigitosCadena "SE 2431 X" == 10

Usar las funciones isDigit y digitToInt

• Definir, por recursión, la función repite :: a -> [a] tal que repite x es la lista infinita cuyos elementos son x.

Pista:

Identifiación de Errores

¿Cuál es el error del siguiente trozo de código en Haskell?

```
Prelude> let a = [ if x < 10 then "BAM! " | x <- [ 5 .. 15 ] ]

<interactive>:36:34: error: parse error on input '|'
```

Ejercicio Final

 Dada una lista de números integrales su tarea consiste en crear una función que imite el resultado de la función map, n veces. La función tiene el siguiente esquema:



Recibirá una función, una lista de números integrales y un número que determinará cuántas veces se debe aplicar la función a cada elemento de la lista. Note que la función a aplicar recibe parámetros integrales y retorna datos del mismo tipo.

Pista:

La función ha retornado 10, 11, 12 puesto que se adicionó 3 veces 3 a cada número de la lista.

La función ha retornado 4, 5, 6 puesto que se consultó el tercer sucesor de cada elemento de la lista.

No se permite el uso de la función map, considere usar una función aparte que aplique una función f sobre un parámetro x, n veces.

• Para la siguiente lista por comprensión al aplicar a cada elemento de la lista 10 veces la función $h(x) = (f \circ g)$ donde

$$f(x) = succ(x) = sucesor de x$$

 $g(x) = 4x$

¿Cuál es el resultado de sumar todos los elementos de la lista retornada por mapNTime