Taller

1. Definir la función areaDeCoronaCircular tal que (areaDeCoronaCircular r1 r2) es el área de una corona circular de radio interior r1 y radio exterior r2. Por ejemplo,

- areaDeCoronaCircular 1 2 == 9.42477796076938

- areaDeCoronaCircular 2 5 == 65.97344572538566

- areaDeCoronaCircular 3 5 == 50.26548245743669

2. Definir la función palindromo tal que (palindromo xs) se verifica si xs es un palíndromo; es decir, es lo mismo leer xs de izquierda a derecha que de derecha a izquierda.

- Por ejemplo,

- palindromo [3,2,5,2,3] == True

- palindromo [3,2,5,6,2,3] == False

3. 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. Por ejemplo,

- triangular 3 4 5 == True

- triangular 30 4 5 == False

- triangular 3 40 5 == False

- triangular 3 4 50 == False

4. Definir por recursión la función potencia :: Integer -> Integer -> Integer tal que (potencia x n) es x elevado al número natural n. Por ejemplo,

- potencia 2 3 == 8

5. Definir, por comprensión, la función sumaDigitosC :: String -> Int tal que (sumaDigitosC xs) es la suma de los dígitos de la cadena xs. Por ejemplo,

- sumaDigitosC "SE 2431 X" == 10

- Nota: Usar las funciones isDigit y digitToInt.

6. Definir, por recursión, la función

- repite :: a -> [a]

- tal que (repite x) es la lista infinita cuyos elementos son x. Por ejemplo,

- repite 5 == [5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,...

- take 3 (repite 5) == [5,5,5]

- Nota: La función repite es equivalente a la función repeat definida en el preludio de Haskell.