Principles for software composition 2020/21
04 - Haskell

[**Ex. 1**] A list is *palindrome* if it is the same when scanned from left to right and from right to left. For example, the strings `"noon"` and `"civic"` are palindrome.
Write a Haskell function `pal` that checks if a list is palindrome.

[**Ex. 2**] Using the function `pal` from Ex. 1, write a Haskell function that takes a list of lists `xxs` and returns the list of palindrome lists in `xxs`.

[**Ex. 3**] Write a Haskell function `select` that takes a list of integers and return the list of elements that are followed by its immediate successor. For example, `select [1,2,5,7,3,4]` must evaluate to `[1,3]`.

[**Ex. 4**] Write a Haskell function `points` that takes a function `f :: Int -> Int` and the extremes of an interval and returns the list of points `(x,f(x))` for all the values in the interval.

[**Ex. 5**] A positive natural number is called *perfect* if it is equal to the sum of its proper positive divisors. For example $6 = 1{+}2{+}3$ and $28 = 1{+}2{+}4{+}7{+}14$ are perfect numbers.
Write some Haskell code to generate the list of all perfect numbers.

[**Ex. 6**] Write some Haskell code that generates the list of Fibonacci numbers.

[**Ex. 7**] Collatz's chains are built as follows: the chain starts with a positive number: if it is 1 we stop; if it is even we continue the chain dividing it by 2; if it is odd we continue the chain by multiplying it by 3 and adding 1. Examples of Collatz's chains are `[3,10,5,16,8,4,2,1]` and
`[7,22,11,34,17,52,26,13,40,20,10,5,16,8,4,2,1]`.
It is conjectured that for all starting numbers the chains finish at the number 1. Write some Haskell code to compute how many Collatz's chains starting with numbers between 1 and 100 have a length greater than 15.

[**Ex. 8**] Define a new Haskell data structure for representing triangles on a cartesian plane and two functions for computing their perimeter and area.
Hint: given the lengths a, b, c of the sides of the triangle and letting $s = \frac{a+b+c}{2}$ the semi-perimeter, you can use Heron's formula to compute the area as the square root of $s(s-a)(s-b)(s-c)$.
Note: Heron's formula as given above is numerically unstable for triangles with a very small angle when using floating point arithmetic.