COMP 141: Haskell — Part 2

Instructions: In this exercise, we are going to review a bunch of Haskell structures.

(1) Use `!!` to define function `second` that returns the second item of the input list.

(2) Use `head` and `tail` to define function `second1` that returns the second item of the input list.

(3) Use `!!` to define function `fourth` that returns the fourth item of the input list.

(4) Use `head` and `tail` to define function `fourth1` that returns the fourth item of the input list.

(5) Define function `secondFromLast` that receives a list and returns the second element from last. Use `!!` infix operator and `length` function.

(6) Define function `secondFromLast1` that receives a list and returns the second element from last. Use `last` and `init` functions.

(7) Use `!!` and `length` to define function `nthFromLast` that receives a number `n` and a list, and returns `n`th item of the list from last. For example, `nthFromLast 3 [1..7]` must return `5`.

(8) Define function `secondHalf` that receives a list, cuts it in half, and returns the second half. For example, if the input is `[1,2,3,4]`, then the output would be `[3,4]`.

(9) Remember that `xs !! n` returns the `n`th element of list `xs`. Let's define our own version of operator `!!`, called `atIndex`, e.g., `atIndex 3 [1,2,3,4,5,6]` must return `4`. Use `take` and `last` functions to define `atIndex` function.

(10) Define function `trimList` that receives a list and removes the first and last elements of that list. For example, if the input is `"abcdef"`, then the output is `"bcde"`. Define the function, using `take` and `drop` functions.

(11) Use `take` and `length` to define function `firstQ` that receives a list as input and returns the first quarter of it. For example, if the input is `[1..20]` then the output would be `[1,2,3,4,5]`.