COMP0002:
Tutorial Questions on Primitive Recursive Functions

This tutorial asks you to write functions in recursive function style.

1. Use recursion over lists to write a function

   `elemNum :: Eq a => a -> [a] -> Integer`

   so that `elemNum x xs` returns the number of times that `x` occurs in the list `xs`.

2. Define a function

   `unique :: [a] -> [a]`

   so that `unique xs` returns a list of elements of xs that occur exactly once in `xs`.

3. Write a property that links `elemNum` and `unique`

4. Write a function

   `insert :: Ord a => a -> [a] -> [a]`

   which correctly inserts an element into a sorted list in ascending order.

5. Write a function that implements insertion sort

   `isort :: Ord a => [a] -> [a]`

   that takes a list and returns a sorted list (and which calls `insert`).

6. Modify the definition of `insert` so that `sort`

   - The list is sorted in descending order
   - duplicates are removed

   e.g. `isort [2,1,4,1,2] = [4,2,1]`