Programming Paradigms Exercise 5 - Haskell 1

Johann Gamper Marco Montali Thomas Tschager

2nd Semester 2016/17

1. Write a program that computes the greatest common divisor of two numbers x and y using the Euclidean algorithm. The Euclidean algorithm is defined as follows: if $x = y$, then return x (or y), otherwise $gcdea(x, y) = gcdea(x - y, y)$ where $x > y$.

2. Write a function `noOfElem` that counts the number of elements in a list. Your function should return the same result as the function `length`.

3. Use the function `noOfElem` of the previous exercise to write a function `countElem` which counts how many times a given element appears in the list. Do not use the function `length`. You may use other functions, though.

4. (a) Write a function that takes two lists x and y as input and returns `True` if x is a prefix of y (otherwise it returns `False`).

   For example,

   - `[]` is a prefix of any list
   - `[3,5]` is a prefix of `[3,5,10,9,8]`
   - `[3,5,7]` is not a prefix of `[3,5]`
   - `[2,3]` is not a prefix of `[1,2,3,4]`
   - non-empty lists are never a prefix of `[]`

   (b) Write a function that takes two lists x and y as input and returns `True` if x is a subsequence of y (otherwise it returns `False`).

   For example,

   - `[]` is a subsequence of any list
   - `[3,5]` is a subsequence of `[3,5,9,8]`
   - `[3,5,7]` is not a subsequence of `[3,5]`
   - `[2,3]` is a subsequence of `[1,2,3,4]`
   - non-empty lists are never a subsequence of `[]`

5. Write a function `orderIt` that sorts a list. Implement the selection sort algorithm. You can use the following functions:

   - `minimum`, which returns the minimum element of a list, e.g. `minimum [3,5,2,4]` would return `2`
   - `delete`, which deletes the first occurrence of an element from a list, e.g. `delete 2 [3,5,2,4]` would return `[3,5,4]`

   In order to use `delete` you have to import the module `Data.List`

   Bonus: Can you also implement the Quicksort algorithm?