CSCI 3675 – Principles of Programming Languages
Fall 2022
Homework 3 – Functional programming with Haskell
Due Sunday, October 2, at 11:59 PM

All the solutions for this assignment should be implemented in Haskell (13.75 pts each). For each problem, clearly indicate the `type` of the function.

1. Write a recursive function `squareOddsRecursive`, that returns the square of each odd number in the list. For example, `squareOddsRecursive [0, 2, 1, 7, 8, 56, 17, 18]` should return `[1, 49, 289]`.

2. Write a function `squareOddsComprehension`, using list comprehension, that returns the square of each odd number in the list. For example, `squareOddsComprehension [0, 2, 1, 7, 8, 56, 17, 18]` should return `[1, 49, 289]`.

3. Write a function `capitalizeComprehension`, which, given a word, will capitalize it. That means that the first character should be made uppercase and any other letters should be made lowercase. For example, `capitalizeComprehension "grEENVIlle"` should return `"Greenville"`. Your definition should use a list comprehension and the functions `toUpper` and `toLower` that change the case of a character (you need to use `import Data.Char` prior to calling these functions).

4. Write a recursive function `prodRecursive`, that returns the product of the numbers in a list (you should not use the `product` function from the standard prelude). For example, `prodRecursive [3, 2, 1]` should return `6`, and `prodRecursive [3.0, 2.0]` should return `6.0`.

5. Write a recursive function `palindromeRecursive`, that returns `True` if the given list is a palindrome (reads the same backward as forward), and `False` otherwise. For example, `palindromeRecursive "abba"` should return `True`, and `palindromeRecursive "abb"` should return `False`. No other functions should be called from this function.

6. Write a recursive function `inRangeRecursive`, that returns all numbers in the input list within the range given by the first two arguments (inclusive). For example, `inRangeRecursive 5 10 [9,3,12]` should return `[9]`.

7. Write a function `inRangeComprehension`, using list comprehension, that returns all numbers in the input list within the range given by the first two arguments (inclusive). For example, `inRangeComprehension 5 10 [9,3,12]` should return `[9]`.

8. Write a recursive function `power a b`, that returns a^b, where b is a non-negative integer (you should not use any of the exponentiation operators: `^`, `^^` and `**`). For example, `power 2.0 1` should return `2.0`, and `power 2 3` should return `8`.

Submission

- Submit your work as one `hwk3.hs` file containing all your functions, via Canvas.
  - The file must be in a simple text format; do not submit Word, PDF, RTF, JPG, etc.
  - Also make sure that any auxiliary information (such as your name or question numbers) is commented out.