Assignment #9
CS 3060 Programming Languages, Spring 2020
Instructor: Dr. Roy

Scala-Haskell

Due Date: May 7 (Thu) @ 11.59 PM.
Total Points: 60 points
Note: This is an extra assignment, which may help you improve your grade as we will be picking up your best 8 assignments out of 9 assignments.

Directions: Using the source provided via Gitlab `https://gitlab.com/sanroy/sp20-cs3060-hw/`, complete the assignment below. The process for completing this assignment should be as follows:

1. You already forked the Repository "sanroy/sp20-cs3060-hw" to a repository "yourId/sp20-cs3060-hw" under your username. If not, do it now.
2. Get a copy of hw9 folder in "sanroy/sp20-cs3060-hw" repository as a hw9 folder in your repository "yourId/sp20-cs3060-hw"
3. Complete the assignment, committing changes to git. Each task code should be in a separate file. As an example, task1.scala for Task 1.
4. Push all commits to your Gitlab repository
5. If you have done yet done so, sdd TA (username: pprabesh for Section 1 and username: prabeshpaudel for Section 2 of CS 3060) as a member of your Gitlab repository

Tasks:

1. **Task #1: (20 points)** Scala. Let's get back to Task 2 of the last Scala assignment where we developed a mini web crawler. You know that in Task 2c we did not use parallel computation (i.e. multi-Threaded processing) last time. Your current task is to do exactly that, i.e., redo Task 2c through parallel computation. You need to use the concept of `par` collection, and concept of functional programming while ensuring that your program is free from `side effect`. *Writing README carries 2 points.*

2. **Task #2: (20 points)** Haskel. Write a Haskell function named `myCount` which takes two parameters, an item `p` and a `list`. The myCount function returns how many times `p` appears in `list`. Your function needs to be generic, e.g. should work for integer as well as characters. Test your function as follows and report the result in README.

   myCount 'a' ['b','a','a','b','a'] should return 3

   myCount 3 [3,1,2,2,3] should return 2

   Writing README carries 2 points.

3. **Task #3: (20 points)** Haskel. Haskell recursive type: Write a function `lessThan10` which takes a tree as input and returns the number of nodes in the tree that hold value less than 10. Use the following data type for the tree, where each internal node as well as each leaf node holds a value.

   data Tree a = Subtrees a [Tree a] | Leaf a deriving (Show)

   Build a tree of 10 or more nodes with height 3 or more (where each node holding an integer) and apply your `lessThan10` function on the tree, and report the output in README file. *Writing README carries 2 points.*