Assignment #6
CS 3060 Programming Languages, Spring 2021
Instructor: S. Roy

Scala #2

Due Date: March 31 @ 11.59 PM.
Total Points: 50 points

Directions: Using the source provided via Gitlab `https://gitlab.com/sanroy/sp21-cs3060-hw/`, complete the assignment below. The process for completing this assignment should be as follows:

1. You already forked the Repository "sanroy/sp21-cs3060-hw" to a repository "yourId/sp21-cs3060-hw" under your username. If not, do it now.
2. Get a copy of hw6 folder in "sanroy/sp21-cs3060-hw" repository as a hw6 folder in your repository "yourId/sp21-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: prabeshpaudel) as a member of your Gitlab repository

Tasks:

1. **(12 points) Task #1:** In this task, your code creates a random list of 40 shape objects, then traverses the list from start to end, and computes the total perimeter of the shape objects. First you need to implement the class hierarchy diagram of the shape types, which is available in the same directory. Shape is an abstract class which has only a "color" attribute whereas Circle class, Square class, and Hexagon class are concrete children of Shape class, and they have more constructors. Note that you do not know beforehand the order of the shape objects (i.e. Circles, Squares, and Hexagons) created in the random list, e.g. you do not know beforehand whether the 1st item is Circle or Square or Hexagon. *Writing README carries 1 point.*

   Note. When you traverse the list to calculate the total perimeter, you need to call the perimeter() function of each shape object.

   Hint: While building the list of shape objects, generate a random number 0 or 1 or 2; if 0, then you may add a Circle object; if 1, then you may add a Square object, else add a Hexagon object to the list.

2. **Task #2a: (6 points)** Write a Scala program which takes a webpage url (say x) from the user (or as a parameter), and then download webpage x. Count the number of images (i.e. "") and scripts (i.e. "<script ...> </script>") present in x. As an example, url x can be "https://www.bbc.com".

   Task #2b: (6 points) Say the webpage of url x contains *links* to other webpages which we denote by y, i.e., y is a set of (zero, one, or more) webpages. Your program needs to download all such webpages (i.e. x as well as y). Count the total number of images on x and y. See Figure 1.

   Task #2c: (5 points) Also, your program needs to count how many webpages (found in Task 2b) have more than 2 images.

   Task #2d: (5 points) Do Task 2b again, but now you are using the `par` (i.e. parallel collection). How much time does the concurrency usage save compared to the serial run in Task 2b?

   Writing README for Task 2 carries 2 points.

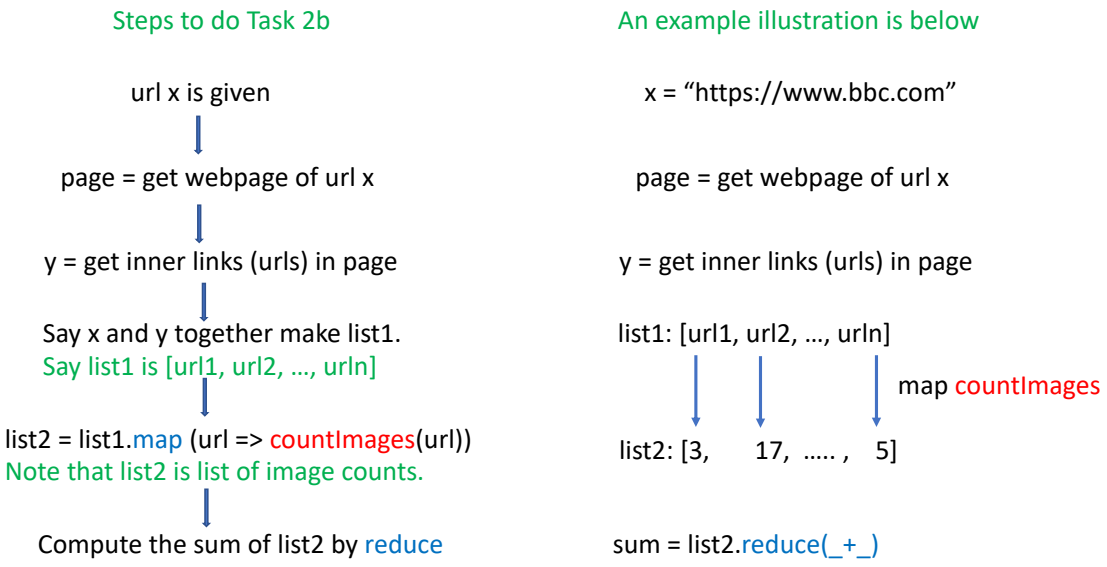

Figure 1: The steps to do for Task 2b.

3. **Task #3a: (8 points)** Write a function foo which takes a list t of strings as input. For each string s in t, foo uses the higher-order function `map` to compute the number of unique characters present in string s. Finally, foo returns a new list containing these numbers. As an example, if t is ["aabcdb", "bcdd"], then foo should return [4, 3].

   Task #3b: (8 points) Write a function foo which takes a list t_1 of integers as input. Function foo needs to construct (and return) a list t_2 where t_2 has the same items as t_1, but pair-wise swapped. As an example, if t_1 is List(3,4,5,6,7,8) then t_2 is List(4,3,6,5,8,7); if the list has odd number of items, then the last element should stay in its place e.g., if input t_1 is List(3,4,5,6,7) then output t_2 is List(4,3,6,5,7).

   Writing README for Task 3 carries 1 point.