Assignment #3
CS 3060 Programming Languages, Fall 2022
Instructor: S. Roy

Scala #1

Due Date: Problems of "hw3a" are due on Oct 15 whereas others on Oct 21.
Total Points: 60 points

Directions: Using the source provided via Gitlab `https://gitlab.com/sanroy/fa22-cs3060-hw/`, complete the assignment below. The process for completing this assignment should be as follows:

1. You already forked the Repository "sanroy/fa22-cs3060-hw" to a repository "yourId/fa22-cs3060-hw" under your username. If not, do it now.
2. Get a copy of hw3 folder in "sanroy/fa22-cs3060-hw" repository as a hw3 folder in your repository "yourId/fa22-cs3060-hw"
3. Complete the assignment, committing changes to git. Each task code should be in a separate file. As an example, task6.scala for Task 6.
4. Push all commits to your Gitlab repository
5. If you have done yet done so, add TA Akhil (username: ayerrab) and Roy (username: sanroy) as a developer of your Gitlab repository

Tasks:

1. **(4 Points. part of hw 3a) Task #1:** Write a Scala program which asks the user to type 3 lines (e.g., before going to the next line the user will hit the 'Enter' key, etc.) on keyboard, and saves the lines to a file named "file.txt". Then, the program opens the same file and counts the number of words and reports the number. Your program file should be named as task1.scala and should have necessary documentation. Also, create a README file, showing one sample running of your program and the output.

2. **(4 Points. part of hw 3a) Task #2:** Write a Scala program which asks the user to type the name of a file. If the file-content (Note: we are NOT talking about the filename string) contains "cpp" or "ruby", then print "The file content is good". If the file-content contains "haskell" or "scala", then print "The file is awesome". Otherwise, print "The file is boring". Your program file should be named as task2.scala and should have necessary documentation. Also, create a README file, showing one sample running of your program and the output.

3. **(4 Points. part of hw 3a) Task #3:** Write a Scala program which prints the string "The square root of x is y" 20 times while substituting x by numbers from 5 to 24 where y is $x^{1/2}$. Your program file should be named as task3.scala and should have necessary documentation. Also, create a README file, showing one sample running of your program and the output.

4. **(7 Points. part of hw 3a) Task #4:** Write a Scala program called $sumOfPower$ to calculate the sum $1^1 + 2^2 + 3^3 + ... + 10^{10}$ without using an exponent operator. You can do this using nested *for* loops. Verify: The sum equals 10405071317. Your program file should be named as task4.scala and should have necessary documentation. Also, create a README file, showing one sample running of your program and the output.

5. **(7 Points. part of hw 3a) Task #5:** Write a function called $doSplit$ which, given a string and a specific character, return a list which is substrings of the original string from one instance of the specific character to the next. Of course, do this without using built-in functions to the extent possible.

   An example: if the given string is pq$xyz$$ab$c and given char is $, then the output should be List("xyz", "", "ab").

   Your program file should be named as task5.scala and should have necessary documentation. Also, create a README file, showing one sample running of your program and the output.

6. **(8 Points.) Task #6:** Write a Scala program to find if a given number is a Kaprekar number. 9 is a Kaprekar number since $9^2 = 81$ and $8 + 1 = 9$

   297 is also Kaprekar number since $297^2 = 88209$ and $88 + 209 = 297$.

   In short, for a Kaprekar number k with n-digits, if you square it and add the right n digits to the left n or n-1 digits, the resultant sum is k.

   Your program file should be named as task6.scala and should have necessary documentation. Also, create a README file, showing one sample running of your program and the output.

7. **(14 Points) Task #7:** Go to `http://www.textfiles.com/stories/` and check that this site [1] hosts multiple stories while each story is in a textfile. Download two textfiles of your choice, which have atleast 600 words, and save the files as `story1.txt` and `story2.txt`. Your program needs to read these files and process them to collect some statistics. In particular, for each story x report the total number of unique words (i.e., without counting repetition) in x. Also, report the third-most frequent word in x and its frequency. Also, find the number of common (and unique) words over these two stories (i.e., if both the stories have a same word w, then we consider that there is one common word w). **Hints:** You may use List, Map (or HashMap), and Set data structures as they are available in Scala. You may design a regular expression to define a *word*. Writing README file carries 2 points.

8. **(12 Points) Task #8:** Write a function foo that takes two lists of integers and returns a list of tuples as explained with the following examples. You are not allowed to use any library function.

   foo(List(1,2,3), List(21, 22, 23)) returns List((1,21), (2,22), (3, 33))

   foo(List(1,2,3), List(21, 23)) returns List((1,21), (2,23))

   foo(List(1,2), List(21, 22, 23)) returns List((1,21), (2,22))

   Note that if the lists have unequal number of items, then foo ignores the extra items in one of the lists. Writing README file carries 2 points.

[1] Disclaimer: we did not really check whether this website contains any improper story or language. If you find something improper, please ignore this site and use some other source