Regular Expressions

SI 206 Homework #6

Due: October 31st

In this homework, you will help Sherlock Holmes find clues to solve a mystery. You have been given a text file 'The_Adventure_of_Sherlock_Holmes.txt' that contains a lot of text with hidden clues in the form of *IP addresses, zip codes, dates, URLs* of certain events in the text. Your task is to use *regular expressions* to find the clues to help Sherlock in his investigation.

To do so, you will complete the following functions in mystery.py:

1. find_IPs (filename)

This function finds and returns all IP addresses from a text file that match a regular expression. IP addresses are IPv4 format, namely four one-to-three digits number separated by dot. (e.g. 192.168.106.123, 127.0.0.1)

2. find_zipcodes (filename)

This function finds and returns all zip codes from a text file that match a regular expression. In the text file, a valid zip code is any of the following formats:
xxxxx (eg, 48109)
xxxxx-xxxx (eg, 48109-0243)

3. find_dates (filename)

This function finds and returns dates from a text file that match a regular expression. You will write the regular expression. A valid date is any date that follows any of the following formats:
mm/dd/yyyy
mm/dd/yy
mm-dd-yyyy
mm-dd-yy
mm.dd.yyyy
mm.dd.yy

Any date that does not follow any of the above formats should not be returned from this function. For example, 12162019 is not a valid date and should not be returned.

4. find_domains (filename)

This function finds *URLs* and returns *their domains* from a text file that match a regular expression. You will write the regular expression.

A URL starts with either *http://* or *https://.* A domain is the part of a URL after *http://* or *https://* (e.g. the domain of *https://google.com* is *google.com*). For the purpose of this assignment, we will define a valid domain as the one that follows the following rules:

1) can contain letters, numbers, and dots
2) does not contain any special characters after *http://* or *https://*

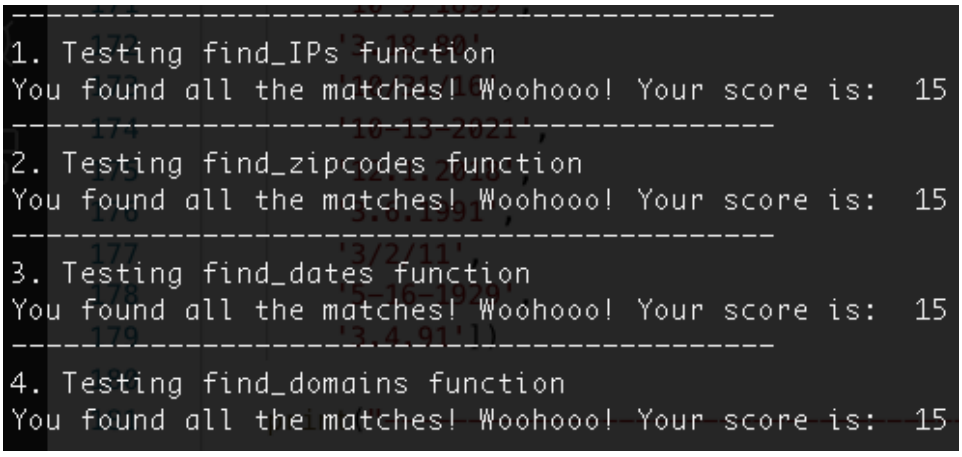
Note: We have included a statistics() function that prints the statistics of your program. It is designed to test your functions and provide you a score based on the number of correct matches. You do not need to change anything in statistics() but you can use its results to understand what you might be missing in your regular expressions.

Grading Rubric (60 points)

This rubric does not show all the ways you can lose points.

15 points for successfully passing all of the tests for find_zipcodes
15 points for successfully passing all of the tests for find_IPs
15 points for successfully passing all of the tests for find_dates
15 points for successfully passing all of the tests for find_domains

Sample Output



Extra Credit (3 points):

Write a function *count_word(filename,word)* to return a count of the number of times a specified word or its plural appears in a file. It should match the word when it starts a sentence also (starts with a capital letter). It should not match any additional letters after the word. For example, if called on “well” it should match “Well”, “well”, "wells", "Wells", but not “farewell”. You MUST use a regular expression to earn credit for this part.

Submission:
Make at least 3 git commits and turn in your GitHub repo URL on Canvas by the due date to receive credit.