**Regular Expressions**

**SI 206 Homework #6**

**Due: March 14th**

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 *phone numbers*, *email addresses*, and *dates* 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\_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 must not be returned.

**2. find\_emails(filename)**

This function finds and returns emails from a text file that match a regular expression. You will write the regular expression.

An email address has the format *localpart@domainpart.* For the purpose of this assignment, we will define a valid email address as the one that follows the following rules:

1. has at least one non-whitespace character, followed by **an at-sign**, followed by at least one more non-whitespace character.
2. does not contain any special characters in the local part or domain part (except .)
3. may contain letters or numbers in the local part as well as in domain part.

Any email that does not match the above criteria should not be returned from this function. For example, “ @2pm” is not a valid email format and must not be returned.

**3. find\_phoneNumbers(filename)**

This function finds and returns all phone numbers from a text file that match a regular expression. You will write the regular expression. In the text file, a valid phone number is any number that has 10 digits and follows any of the following formats:

xxx/xxx-xxxx (eg, 206/782-8410)

xxx.xxx.xxxx (eg, 206.782.8410)

xxx xxx xxxx (eg, 206 782 8410)

(xxx) xxx-xxxx (eg, (206) 782-8410)

xxx-xxx-xxxx (eg, 206-782-8410 )

xxxxxxxxxx (eg, 2067828410)

Any number that does not have either 10 digits or does not follow one of the above formats is not a valid number. For example, 206-7828-8410 is not a valid phone number and should not be returned.

*Hint*: start writing regular expression for easy formats first like 206 782 8410, 206.782.8410, etc. and then try to extend your regular expression to match the harder ones like

(206) 782-8410.

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)**

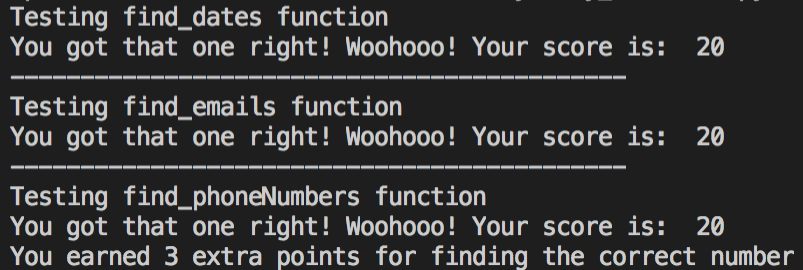
This rubric does not show all the ways you can lose points.

20 points for successfully passing all of the tests for find\_phoneNumbers

20 points for successfully passing all of the the tests for find\_emails

20 points for successfully passing all of the the tests for find\_dates

**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.