Assignment 1

Submit by: Friday 4th Feb- 2022 (11:59 PM-midnight)
Points: 7
This is an individual assignment

1. Word List Statistics (2pt)

Read Chapter 9 Case Study: Word Play in Think Python 2e. Write a program that reads the word list from the text file words.txt available at http://thinkpython2.com/code/words.txt (uploaded on D2L) and prints the following output:

- Number of words in the list
- Average word length
- Maximum word length
- Number of words that use all vowels 'aeiou'
- Number of abecedarian words

Use functions as appropriate. Document your code and present your findings, e.g. screenshot of output.

2. Text Analysis (2pt)

According to Wikipedia, The Feynman Lectures on Physics are likely the most popular physics books ever written. We would like to understand better why that is. Maybe it is related to how the text is structured. The goal of this exercise is to analyze the text in Chapter 1.1 Introduction of the Feynman Lectures available here http://www.feynmanlectures.caltech.edu/I_01.html

Design a program that reads this chapter from a text file, uses functions where appropriate, and prints the following information:

- Number of words in the text
- Number of sentences in the text
- The average number of words per sentence
- The average number of sentence parts (delineated by commas) per sentence
- Number of personal pronouns, e.g. I, me, my, you, we

Document your code and present your findings, e.g. screenshot of the output.

3. Credit card number check (1pt)

You have been hired by MasterCard to write a function which checks if a given credit card number is valid. Your function check(card_number) should take a string card_number as input.

First, if the string does not follow the format "dddd dddd dddd dddd", where each d is a digit, it should return False.

Second, if the sum of the digits is divisible by 10 (a "checksum" method), then the procedure should return True, otherwise it should return False.

For example, if card_number is the string "9384 3495 3297 0123" then although the format is correct, the digit's sum is 72 so you should return False.

Implement the function and write a program to test your function. Document your code and demonstrate proper implementation using positive and negative test cases.

Test input:

- card_number = "9384 3495 3297 0121"
- card_number = "9384 3495 3297 0123"
- card_number = "9384 3495 3297 01236"
- card_number = "9384 3495 3297 012A"

4. Parsing industrial input file by using regular expression (2pt)

A regular expression is written in a very specific language, and its purpose is to define a text pattern. One of the biggest uses of regular expressions is to detect specific text patterns within a larger text file, such as a log file. For example, you might write a regex to look for the specific text that represents an HTTP 500 error in a web server log file or to look for email addresses in an SMTP server log file. In this assignment your task is to parse an industrial benchmark and extract the following patterns:

- Read bench.txt file as input.
- Extract all the rows with the following pattern from the text file (by using regex):
    - Pattern:
        - - inst[digit_number] [MACRO NAME] + PLACED ([X] [Y]) [DIRECTION] ;
    - For example:
        - - inst2591 NAND4X2 + PLACED (100000 71820) N ;
            - In this example:
                - Digit_number: 2591
                - MACRO Name: NAND4X2
                - X: 100000
                - Y: 71820
                - Direction: N
- Traverse all instances and print on the console.

Expected output:

- inst2015 NAND3X2 + PLACED (88000 78660) N ;

```
- inst2591 NAND4X2 + PLACED ( 100000 71820 ) N ;
- inst2908 OR4X1 + PLACED ( 85600 75240 ) FS ;
- inst3428 BUFX3 + PLACED ( 86000 82080 ) FS ;
- inst3502 NOR4X2 + PLACED ( 92800 75240 ) FS ;
- inst3444 BUFX3 + PLACED ( 96800 82080 ) FS ;
- inst4132 NAND4X1 + PLACED ( 102000 85500 ) N ;
- inst4183 NOR4X4 + PLACED ( 96800 75240 ) FS ;
- inst4062 AOI222X1 + PLACED ( 96000 85500 ) N ;
- inst4678 NOR2X1 + PLACED ( 90800 82080 ) FS ;
- inst4189 AOI22X2 + PLACED ( 85200 71820 ) N ;
- inst4597 NAND4X1 + PLACED ( 94400 71820 ) N ;
- inst4382 NAND4X1 + PLACED ( 84000 78660 ) N ;
- inst5333 AOI221X1 + PLACED ( 89200 85500 ) N ;
- inst5638 BUFX6 + PLACED ( 96000 78660 ) N ;
- inst6286 AO22XL + PLACED ( 100000 78660 ) N ;
- inst5275 AOI22X1 + PLACED ( 89600 75240 ) FS ;
- inst5821 NAND3X1 + PLACED ( 84400 85500 ) N ;
- inst6458 AOI221X2 + PLACED ( 99200 82080 ) FS ;
- inst6050 NAND4X1 + PLACED ( 97200 71820 ) N ;
- inst5195 AOI221X1 + PLACED ( 89600 71820 ) N ;
- inst7234 AO22XL + PLACED ( 93200 82080 ) FS ;
```

What to hand in

Upload four python files and a document with your results to D2L dropbox with solutions to above exercises:

1. A01_1_stats.py
2. A01_1_avoids.py
3. A01_2_feynman.py
4. A01_3_creditcard.py
6. A01_parse.py
7. A01_results.pdf [screen shots of all results]