H2 Computing Practical Worksheet 3

Set 1 – Please attempt the following questions by implementing the solutions using Python 3.

1 Write code to generate all the positive integers up to `n` that are perfect squares and save them in a text file on your desktop called "`perfect_squares.txt`".

Note: store one integer on each line in the file.

2 Write code to generate all the positive integers from `n + 1` to `n + m`, and then append them to the file used in question 1.

Note, do not overwrite the perfect squares from Question 1.

3 Download the file: https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data

Write code to read and store the contents of the file. When storing the data, you should use a list of tuples, where each tuple in the list corresponds to 1 line of data from the file. More specifically, the tuple should correspond to:

```
(FLOAT, FLOAT, FLOAT, FLOAT, STRING)
```

Set 2 – Please attempt the following questions by implementing the solutions using Python 3.

4 Download the file at: http://itu.dk/people/pagh/sad12/primes1000000.txt

Write a function that refers to the above file to test if a given positive integer is a prime number. If the given numbers exceeds 1,000,000, the function should still utilise the file to provide an efficient solution. Determine a good upper limit on the input.

5 Download the file at:
http://www.textfiles.com/etext/AUTHORS/SHAKESPEARE/shakespeare-romeo-48.txt

Write code that finds the frequencies of each unique word in the file and saves this list in a file named "R_&_J_Word_Frequencies.txt".

Your code should adhere to the following when processing these frequencies:

- You must remove all punctuation before you begin processing
- All frequencies should ignore case (i.e., if two words are only different in terms of the case of certain letters, then they are both to be treated as the same word – e.g., "problem" and "PROBLEM" and "PrObLeM" are all to be considered the same word)
- Each word corresponds to a string bordered by a space of an end line character

Note, the file you save should contain n lines, where n is the number of unique words. On each line we should see <word>,<frequency>.

Additional References

- “How to Think Like a Computer Scientist” Chapters 11.1 – 11.3: http://www.greenteapress.com/thinkpython/thinkCSpy/html/chap11.html

- “Python Programming” Section on Files: https://en.wikibooks.org/wiki/Python_Programming/Files

There are many other exceptional resources available on the internet. If you find any, please share with the class via group chat.