Programming Exercise

Meant only to improve your proficiency in this subject.
These programming assignments will not be evaluated/graded

1. Consider the text available in this website

   http://shakespeare.mit.edu/allswell/full.html- to answer the following questions

   (a) Find the frequency the word "BERTRAM" (all in caps, ignore font types such as bold, italics, etc.) that appears at the start of the sentence?

   (b) Is the ratio of the frequency of $\frac{BERTRAN}{Bertram} > 0$

   (c) What is the average sentence length of this document?

   (d) What is the vocabulary (unique set of words) size? Ignore alphanumeric and numeric terms, if available

   (e) Find the total count of the words that end with the exclamation mark(!)

2. Construct the binary incidence matrix using the features extracted from the corpus. The corpus (271 text documents) is available at

   https://github.com/Ramaseshanr/anlp/blob/master/corpus/phy_corpus.txt. It contains contains questions from Kinematics class of physics problems sourced from the Internet

   - In this assignment, you need to develop a python program that uses the knowledge related to Kinematics and build a table similar to the one shown below for all the documents in the corpus.
   - The program should be able to read each problem, capture the known values (such as speed=10m/s, time=5s) and fill the respective cells in the table. For example, if you find 10 m/s for document 1, fill the speed with value row for D1 as 1.
   - Please note that problems may or may not contain all nine terms listed.
   - The corpus may contain duplicate entries
   - You may use any NLTK or any equivalent APIs for this assignment

Terms	D_1	D_2	...	D_{271}
Speed with value	1	0	...	0
Distance with value	0	0	...	1
Acceleration with value	0	0	...	0
Time with value	0	1	...	0

3. Write a program to find out whether Mandelbrot's approximation really provides a better fit than Zipf's empirical law. Use the same corpus for Zipf and Mandelbrot approximation

4. Write a program to implement Heap's law and find out the prediction of vocabulary in any corpus. Find out whether it is closer to the actual the size of the vocabulary of the same corpus. What could be approximate value for k for the chosen corpus?