Project 1

Due 11:59pm Friday, February 9th. (14 days)

There will be several projects in this class requiring you to do some programming. I recommend the Python programming language, not only is it a smooth transition from C (which many of you know) but I have seen it as a requirement for many job applications, it would be a good thing to add to your resume upon the completion of this course.

Task description

On blackboard you will find the complete text of "War and Peace" I would like you to do the following tasks:

0. Print on the screen total number of words and sentences in the text.

1. Give a number of occurrences of each word and the frequency (number_of_occurrences / total_number_of_words) for this word. To do this you will have to first remove special characters/convert everything to lowercase and calculate the total number of words. I would use dictionaries that map each word to a value that you can then increment. There are might be some other ways to do it.

As an output I would like to see a sorted list in csv format (as a separate csv file with name *result.csv*) of the most frequent word first followed by the second most frequent as well as the number of times that word appears and its overall % of words, so the words “a” “I” “an” should all appear frequently. Make sure that you don’t include spaces (“ “) and that they are filtered out. Look at your csv output to make sure it makes sense. Another possible problem area is apostrophes. You should filter “that’s” to “thats”, otherwise you will very likely get “s” as a very popular word.

Sample CSV output:

```
a,500,0.3
i,300,0.23
the,150,0.12
...
```

2. Give me the number of sentences and the frequency in the novel where the first word in the sentence is “the”, or “The” You will have to do some string comparisons (look for the “.” Or “?” or “!” Signaling the end of a sentence)

3. Find the most frequent two word combination in the text (e.g. “I am” or “synchrotron radiation”). Print it on the screen.

Program specifications:

4. It should print results of tasks 2 and 3 on the screen.

Program specifications:

1. Source code with comments has to be named **project1.py** (NOT myproject.py, prog1.py or anything else)

2. It should read the file text.txt within the same directory and produce CSV file **result.csv** within the same directory.

3. The program should not ask me any questions: 1) the program should process the file; 2) output required information on the screen; 3) output required information into the file; 4) and exit.

Note: Don't hardcode the path to the file into your program. I may run it in **c:/python** or in **/home/user/students/projects** or somewhere else. Your program should work anyway because the test file will be kept next to it.

3. Your program should run with no errors on python 2.7. (if it doesn't run you lose most of the grade)

4. It should not take more than **1 minute** to produce the result.

Submission:

Make sure your code is commented and you have a clear understanding of what you did.

e-mail the following to eece480f@gmail.com with the subject line "Your name, Project 1":

1. source code with comments (.py).

FAQ

Q1: How in depth does our special character filtering have to be?

A1: Situations occurring very often such as "it's" "don't" should be recognized an properly handled.
First thousand (1000) words by frequency should not contain any non-words such as "s" or "t"

Q2: How to handle hyphens.
A2: Hyphens that split the words in the end of the sentence should be handled.
Hyphens in composite words such as state-of-the-art should be handled if it happens in the middle of the line.
If composite word is in the end of the line e.g.
..... blah blah blah state-of-
the art blah blah .....

you can ignore it or come up with some way to handle it. Make note in the code what did you decide to do.

Q3: How to handle nested sentences?
A3: Sentences such as:

General said: "Brace yourself! Winter is coming!"

Can be counted as one two or three sentences. Decide which way you want to handle it and document it in the code comments. Keep it consistent.