22C:16 Homework 8

Due via ICON on Wednesday, March 30th, 4:59 pm

Submit the solutions to all 4 problems, but we will grade some 2 problems of our choice from your submission.

1. I have made available electronic versions of 6 famous novels that I downloaded from Project Gutenberg. Write a program that reads from all of these files and produces as output a file called `dictionary.txt` that contains all of the words that appear in these files. While words may appear multiple times in input files, each word should appear exactly once in `dictionary.txt`. Furthermore, `dictionary.txt` should contain the words in a simple format: one word per line, the words appearing in alphabetical order, and all words in lower case. Use the definition of *word* that we have employed in other recent homeworks: a word is a contiguous sequence of letters (upper case or lower case) such that the character just before is a non-letter and the character just after is a non-letter.

   Since the downloaded files have words spelled correctly, the file `dictionary.txt` can be viewed as a dictionary of words that can be used for other purposes such as for spell checking (see Problem 2). Because of the way we have defined a word, it is possible that items that are clearly not valid English words will make their way into `dictionary.txt`. For example, the contraction `we've` will cause `we` and `ve` to be added to the dictionary. This is okay and your program does not have to do anything to fix this.

2. Write a simple spell-checking program that reads an input file called `input.txt` and outputs words that are incorrectly spelled. To determine if a word is incorrectly spelled, your program should consult the dictionary file constructed in Problem 1.

3. Write a program that reads a bunch of points in 2-dimensional Euclidean space and outputs all the point-triples that are collinear. Exactly as in the previous homework, each point is specified in a separate line with the x-coordinate specified first followed by the y-coordinate. The two coordinates are separated by one or more whitespaces. After all the points have been provided the user will type an extra enter (i.e., an empty line) to indicate that she is done. In general, the coordinates will be floating point numbers. An example of the interaction between your program and the user is given below.

   ```
   Please input the points:
   1.0 1.0
   11 18
   2 2
   18 11
   3.0 3.0

   The following point-triples are collinear:
   (1.0, 1.0), (2.0, 2.0), (3.0, 3.0)
   ```

   In the previous homework (Homework 7, Problem 1) you had to generate all pairs of points. Here you have to generate all triples and you can use you code from that problem and modify it a bit to solve this problem.

   It is possible that there is no collinear triple in the input set of points. In this case, output a message saying so. Finally, you may recall from middle school math that three points are collinear if they lie on the same line. Computing and comparing slopes might be the easiest way to determine collinearity.

4. You should use the file `dictionary.txt` for this problem. Write a program that reads a *partially specified* word and outputs *all* words in `dictionary.txt` that *match* the partially specified word. A partially specified word is one that is missing some letters, with these missing letters indicated by the underscore character. For example, `b__t` is a partially specified word and examples of words that this matches might be `boot`, `best`, `bolt`, etc.
