22C:16 Homework 5

Due via ICON on Wednesday, March 7th, 4:59 pm

What to submit: Your submission for this homework will consist of five files: (i) `gradeDistribution.py` for Problem 1, (ii) `areCollinear.py` for Problem 2, (iii) `collinearityTest.py` for Problem 3, (iv) `processPoints.py` for Problem 4, and (v) `wordDistribution.py` for Problem 5. Each of these Python files should start also with your name, section number and student ID appearing at the top of the file as Python comments. Also make sure that your Python code is appropriately commented. You will get no credit for this homework if your files are named differently, have a different format, or if your files are missing your information.

1. Define a function called `gradeDistribution` that takes a list of exam scores and returns a list that contains the *distribution* of these scores. To be more precise let us suppose that the exam scores are out of 100 and therefore these are floating point numbers in the range 0 through 100 (inclusive of 0 and 100). The distribution of the scores I want you to compute is the number of scores that are in each of the ranges $[0, 10]$, $(10, 20]$, $(20, 30]$, $(30, 40]$, $(40, 50]$, $(50, 60]$, $(60, 70]$, $(70, 80]$, $(80, 90]$, and $(90, 100]$[1]. In other words, the first element in the list returned by your function should be the number of scores in the range $[0, 10]$, the second element should be the number of scores in the range $(10, 20]$, etc. You should use the following function header:

   ```
   def gradeDistribution(examScores):
   ```

   Submit a file called `gradeDistribution.py` containing this function.

2. Define a function that takes as parameters three points in 2-dimensional space and determines if the points are *collinear* (i.e., lie on a straight line). Each point will be represented by a list of length 2 consisting of floating point numbers. The first element in such a list will be the x-coordinate of the point and the second element will be the y-coordinate of the point. The function should return the boolean value `True` if the points are collinear and `False` otherwise. Use the following function header:

   ```
   def areCollinear(p1, p2, p3):
   ```

   Note: To check if points p_1, p_2, and p_3 lie on the same line, you should find the slope of the line passing through p_1 and p_2 and then the slope of the line passing through p_2 and p_3. If these two slopes are identical, then the three points are collinear; otherwise not.

   Submit a file called `areCollinear.py` containing this function.

3. Define a function that takes as parameter a list of points and checks if the list of points contains three that are collinear. If the function finds three points that are collinear, it returns these in a list. Otherwise, the function returns the empty list. Use the following function header:

   ```
   def collinearityTest(pointList):
   ```

   This function should repeatedly call the function `areCollinear`. Submit a file called `collinearityTest.py` containing this function along with the function `areCollinear`.

4. Write a program that reads in a bunch of points in 2-dimensional Euclidean space and outputs either a message saying that the list contains no collinear triple (i.e., a collection of three points) or a message listing three points in the list that are collinear.

   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 blanks. After all the points have been provided the user will type an extra `enter` (i.e., an empty line) to

[1] I am using $(A, B]$ to denote the range $A < s \leq B$ and $[A, B]$ to denote $A \leq s \leq B$.

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:
3.5  -4
2 2
12.0 9.86
1.75 3

The points (3.5, -4), (2, 2), and (1.75, 3) are collinear.
```

This program should call the function `collinearityTest`.

Submit a file called `processPoints.py` containing this program along with the functions `collinearityTest` and `areCollinear`.

5. Write a program that reads some text and produces as output the distribution of words that start with different letters. As in the previous problem, the end of the text is specified by an extra `enter`. Below is an example of how a user might interact with this program. The first line is the prompt produced by the program. Following this prompt, I entered two lines of text and then typed an extra `enter`. The output is a list of 26 numbers. The first number in the list is the number of words in the input that start with the letter `a` or `A`, the second number is the number of words that start with `b` or `B`, etc.

```
Please type your text. Type an extra enter when you are done.
This is a test.
This is a test Also.

[3, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0]
```

Submit a file called `wordDistribution.py` containing this program.

6. **Extra Credit: 10 points.** The 2012 University of Iowa Computing Conference is being held on March 2nd (starting at 6 pm on Friday) and March 3rd. For details visit `http://acm.uiowa.edu/uicc/` The keynote talk is by Prof. David Forsyth, who will talk about inserting synthetic objects into an existing photograph. There are 4 talks on Saturday. Go to the UICC website for details on times and locations.

   Attend any one talk and write a 1 paragraph (5-7 sentences) summary of the talk.
