CS/IT 102: Lab 10 – Files (72 points)

Instructions: In this lab you will be working on reading from and writing to files. Your submitted code file must include the solutions to all problems below. Comment out functions and code as needed during testing. The data files may be found on Piazza/Resources in the Labs section

Write code for each of the functions below. You should use a single main() function to test your functions. You must ensure that all files are correctly opened and closed.

- **Change partners every problem.**
- **Place your functions at the end of this document.**
- **Your code must comply to the problem requirements**
- **Grading:**
  - Problems 1-11: 5 points each
  - Problem 12: 12 points
  - Challenger: 5 extra credit points to account for exam 3
  - Code commenting & variables/functions naming: 5 points

Problem 1: Write a function called `display_file` that takes the name of a file containing integers, and prints the file, one line at a time. Use the `numbers.txt` file to test your code.

Problem 2: Write a function called `display_head` that takes the name of a file and a number n representing the number of lines to display and prints the top n lines in the file. You may test your function with either the `numbers.txt` or `names.txt` files.

Problem 3: Write a function called `print_line_numbers` that takes the name of a file and prints the contents of the file with each line preceded by the line number. For example, if the file contains:

```
Hello
Goodbye
```

the function should display:

```
1  Hello
2  Goodbye
```

Problem 4: Write a function called `count_lines` that takes a file name and returns the number of lines in the file.

Problem 5: Write a function called `read_sum` that takes the name of a file containing integers and returns the sum of all of the integers in the file. You may test your function using the `numbers.txt` file.

Problem 6: Write a function called `read_average` that takes the name of a file containing integers and returns the average of the integers in the file. You may test your function using the `numbers.txt` file.

Problem 7: Write a function called `write_random` that takes the name of a file and a number of numbers to write to the file. The function should write the number of random numbers between 1 and 500 to the file, one integer per line. Verify that your code works by opening the file using notepad++ or other text editor.

Problem 8: Write a function called `read_random` that takes the name of a file. The function should read the numbers from the file and print one integer per line. Your function should return the total of all numbers in the file. Verify that your code works by opening the file using notepad++ or other text editor.

Problem 9: Write a function called `write_ages` that takes the name of a file. The function should prompt the user to enter a name and an age and write those to the file, one name/age pair per line. It should prompt the user if they would like to enter another age. Sample prompts might be:

```
Enter the following data:
Name: Dino
Age: 4
Another record (y/n)?n
```

Verify that your code works by opening the file using notepad++ or other text editor.

Problem 10 Write a function called `read_ages` that takes the name of a file that contains names and ages, one pair per line. The function should print out names and ages one per line. Sample output might be:

```
Name: Dino
Age: 4
```

Use the file you created for problem 9 to test your function.

Problem 11: Write a function called `average_num_words` that takes a file name and returns the average number of words per line in the file. Test your function on the files `lyrics.txt`.

Problem 12: Find the file titled `titanic.csv`. This file contains information about the passengers on the titanic including the name, class, age, gender, and whether or not the person survived (1 = survived, 0 = did not survive). Write a function called `num_survived` that takes a file of titanic information and returns the number of survivors of the sinking.

- Write a function called `num_gender_survived` thats takes a file name and a string representing the gender ("male" or "female") and returns the number of that gender that survived.
- Write a function called `average_age` that takes a file name and returns the average age of the survivors.
- Write a function called `class_survivor` that takes a file name and returns the class with the greatest number of people who survived.

Challenger Starbucks Problem: Find the file titled `starbucks.csv`. That file contains the longitude, latitude, name, and address (in that order) of every Starbucks in the United States. You will write a function that accepts a latitude and longitude as arguments, and prints the address of the nearest Starbucks to that location. Call this function `starbucks()`.

To determine the distance between two pairs of latitude and longitude coordinates, use the following formula. Assume the coordinates are (lat1, long1) and (lat2, long2).

Note: The sine and cosine functions in the Python math module assume radians, but latitudes and longitudes are expressed in degrees. You should use the `math.radians(x)` function to convert from degrees to radians before calculating the sine or cosine.

Formula:

$X = \cos(lat2) * \cos(long2) - \cos(lat1) * \cos(long1)$

$Y = \cos(lat2) * \sin(long2) - \cos(lat1) * \sin(long1)$

$Z = \sin(lat2) - \sin(lat1)$

$Distance = 3958.761 * \sqrt{X^2 + Y^2 + Z^2}$

Test your code using the sample latitudes and longitudes are listed below.

Location	**Latitude**	**Longitude**
Western New England Univ.	42.1155	-72.5197
Fenway Park	42.3464	-71.0975
Empire State Building	40.7484	-73.9857
White House	38.8978	-77.036