Due by 11:59 pm on the date specified in Canvas; 50 points total.

- *Late submissions are penalized 5 points per day late.*

Instructions:

- Complete each of the following problems as a separate Python (.py) file.
- The name of each Python file should be of the form "Lab5_probX.py".
  - Example: Lab5_prob1.py for problem 1, Lab5_prob2.py for problem 2, etc.
- "Zip" all three problem files into ONE zip file and submit to Canvas by the due date.
- Follow variable naming rules as described on pages 43-44.
- Comment frequently in your code using the "# comment" convention described on pages 39-40. At a minimum, you should have a comment line at the beginning of the program with your name and what the program will do. Comments are for the programmer to see. *Be sure comments are not so long that they get cut off on the right side of the monitor (continue to the next line, if necessary, to avoid this situation).*
- Display 1-2 print statements to the user on the purpose of the program before asking for any information. This is for the user to see. *Be sure this display is not so long that it gets cut off on the right side of the monitor (continue to the next line, if necessary, to avoid this situation).*
- Check the accuracy of the output you get out of your code. In other words, how do you know that the output is correct?
- The grading rubric is provided at the end of this document for your reference. Use it to 'check' your work before submitting it.
- **Make sure to call a main function in each problem below in addition to whatever other functions are specified in the problem.**

For each problem in this assignment, be sure to follow the bulleted items in the instructions above (comments to the programmer throughout the program, display purpose statement to the user, etc) in addition to the specific problem instructions.

1. **Problem #1 – Lucas numbers recursion (Chapter 12; 10 points)**
   - Look up "Lucas number" in Wikipedia and compare the definition of computing a Lucas number to the definition of computing a Fibonacci number.
   - Download the fibonacci.py program from Canvas, rename it Lab5_prob1.py, and modify the recursive definition of a Fibonacci number to a Lucas number. Rename the function to emphasize that it is computing a Lucas number and not a Fibonacci number.
   - Tell the user what a Lucas number is (how to compute it) and add in a prompt to the user on how many Lucas numbers to display. Modify the program to display that many Lucas numbers to the user.
2. **Problem #2 – Write Directory Information to a directory.txt file (Chapter 6; 15 points)**
   - A local community group held a 5K run/walk to collect monetary donations for COVID-19 assistance. To show its appreciation, the group is going to host a "desserts event" to thank all its contributors. A representative from the group has contacted you (a skilled Python programmer) to create a directory of contributors. This program will ask the user "How many contributors?" and proceed to prompt the user for each contributor's name, favorite dessert, zip code, and monetary contribution amount. Write a program that writes this information to a text file called "directory.txt."
   - Use an intro() function to display the purpose of this program to the user.
   - Use a "for" loop to repeat prompting the user for the directory information for each contributor.

3. **Problem #3 – Read/Display directory.txt file with Summary Statistics and Exception Handling (Chapter 6; 25 points)**
   - Write a program that reads and displays the directory information stored in the "directory.txt" file from problem #2. Each field should be labeled.

     Sample Output:

     ```
     Name:  Ima Runner
     Favorite Dessert: Apple Pie
     Zip code: 12345
     Contribution amount: $ 611.00
         <<<<Blank line between entries >>>>
     Name:  Stu Dent
     Favorite Dessert: Brownies
     Zip code: 12346
     Contribution amount: $ 600.72
         <<<<Blank line >>>>
         <<<<Blank line >>>>
     Summary Statistics:
     Total Contributions: $ 1,211.72
     Number of Contributors: 2
     Average Contribution Amount: $ 605.86
     ```

   - Display summary statistics as shown above in the sample output with total contributions ($ amount), number of contributors (include a "counter" in your loop), and a calculated average contribution amount. Include units and include the ability for dollar amounts to include the comma for separating by thousands.
   - Use an intro() function to display the purpose of this program to the user.
   - Use a while loop to read the directory.txt contents and to detect the end of the text file. Logic to set this up is on page 306 with examples on pages 307 and 314.
   - Include your code in a try/except block to handle any ValueError or IOError exceptions. I will be checking for the error handling by introducing the error types indicated and seeing if your program can handle it.

Grading Rubric

1. **Problem #1 – Lucas numbers recursion (10 points)**

____ (5) Statement displays to the user the purpose of the program (intro function). Comments to the programmer are included throughout code. Main program logic is included in the main function.

____ (5) Recursive definition is modified to give Lucas numbers. Program is modified to prompt the user for how many Lucas numbers to display and the program provides this many numbers when executed.

2. **Problem #2 – Write Directory Information to a directory.txt file (15 points)**

____ (5) Statement displays to the user the purpose of the program (intro function). Comments to the programmer are included throughout code. Main program logic is included in the main function.

____ (5) Program prompts the user for how many contributors to save in directory.txt file. Uses "for" loop to ask for name, dessert, zip code, and contribution amount.

____ (5) Program uses open/write/close methods to create a directory.txt file and writes to it. Text file can be opened in notepad to verify the existence of the inputted data.

3. **Problem #3 – Read/Display directory.txt file with Summary Statistics and Exception Handling (25 points)**

____ (5) Statement displays to the user the purpose of the program (intro function). Comments to the programmer are included throughout the code. Main program logic is included in the main function.

____ (6) Program uses a "while" loop to read in and detect end of file. Uses readline methods to read in each piece of data. Loop contains a counter to count the number of contributors and a counter to total the monetary contributions.

____ (5) Contents of the file are displayed to the user with labels and formatting as specified in the problem description. See sample output.

____ (4) Summary statistics calculates correctly and displays as shown in the sample output.

____ (5) Handles ValueError and IO Error exceptions that the instructor gives it.

Total: __________ / 50