CSC 401 ASSIGNMENT THREE

Due Date: Tuesday, Sep. 26th by 11:58 PM

The purpose of this assignment is to assess your understanding of

- User defined functions and parameter passing (Section 3.3 and 3.5)
- List methods (Section 2.3 pages 31-33)
- String formatting (Section 4.1 and 4.2)
- Iteration Structure – for loop (Section 3.2)
- Reading files (Section 4.3)

SUBMISSION

- **Include your full name as a comment in the first line of your Python program**
- **Include the problem number as a comment in the second line of our Python program**
- Save each program to a separate file labeled as YourName_hw3_1.py, YourName_hw3_2.py, YourName_hw3_3.py accordingly
- Upload each file to Submissions folder in D2L.

PROBLEMS

Note: you may not use Python statements, functions, data types, etc. that were not discussed in the reading assignment or the lecture notes/videos for this week or previous weeks. This is a class for students who have not programmed before and I expect everyone to code on the same level. If you have a better way of writing the code, then upload two versions: one that codes according to the specifications and the other that demonstrates advanced programming techniques.

I encourage you to use computational thinking to solve the problems. These are straight-forward solutions, but developing a good habit of analyzing the problem and describing the steps will serve you well as the problems get more complex.

1. (function, list, decision, iteration (for loop), formatting 20 points) Write a Python program that separates a list of at least 20 positive and negative integers into a sorted list of even integers and a sorted list of odd integers. There should be two functions in this program:

- evenOdd (lst, EorO) has 2 arguments; lst (list of integers) and EorO a string that contains either an E or O. If EorO has the value E, then return the list of sorted even integer. If EorO has the value O, then return the list of sorted odd integers.
    - Even integers are divisible by 2; odd integers are not divisible by 2.
- main() has no arguments
    - Create the list of at least 20 positive and negative integers.

- Ask the user if they want the list of even or odd integers by entering an E or O, respectively.
- Calls the evenOdd() function passing the list and the response to the above.
- Evaluates the returned results and displays the list in two formatted columns.
- If the user's request does not have any numbers in it, then display 'No even numbers' or 'No odd numbers' respectively.

You must pass arguments from the calling program function to the called program function. (you may not use global variables i.e. variables defined outside of either function.)

Use three test cases: a list with even and odd integers
a list with only even integers
a list with only odd integers

Sample:

```
>>> main()
What do you want displayed, the even (type E) or the odd (type O) numbers: E
The even numbers are:
-200   -2
  10   12
  36   40
  86  146
 198
>>> main()
What do you want displayed, the even (type E) or the odd (type O) numbers: O
The odd numbers are:
-225  -63
  -9   -7
   1    7
  23   39
  45   55
  75   95
```

2. (Function, Iteration (for loop), 15 points) Write a function findFib(n), that will find and return the nth Fibonacci number.

Consider the following sequence of numbers: 1, 1, 2, 3, 5, 8, 13, 21, 34, . . .
The n^{th} number, a_n, of this sequence is given by: $a_n = a_{n-1} + a_{n-2}$

Thus: $a_3 = a_2 + a_1 = 1+ 1 = 2$
$a_4 = a_3 + a_2 = 3 + 2 = 5$
and so on.

Such a sequence is called a Fibonacci sequence. The 3^{rd} number in the sequence is 2, the 4^{th} number in the sequence is 3, and the 5^{th} number in the sequence is 5 and so on.

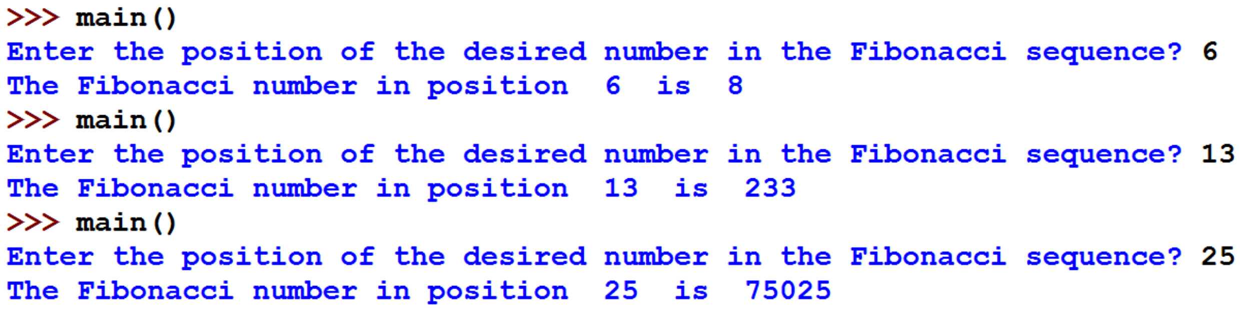
Write a function that determines the n^{th} Fibonacci number given the first two numbers.
There should be two functions in this program:

- findFib(n) has one argument **n** and finds the Fibonacci number as described above.

- main() requests the position of the Fibonacci number from the user, calls findFib(n) passing the number the user requested to the findFib function, and prints the Fibonacci number returned by the findFib function.

Hint: you need to calculate the sequence up to the requested position. It is not required that you use a list to store the sequence, but you may find it useful.

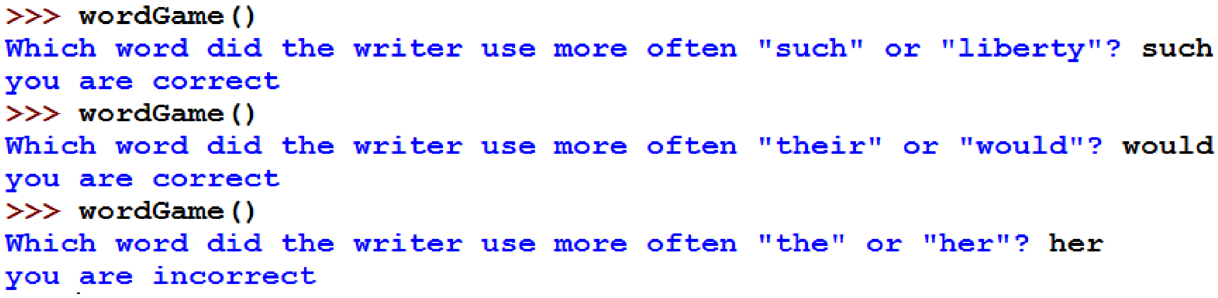
Sample:



```
>>> main()
Enter the position of the desired number in the Fibonacci sequence? 6
The Fibonacci number in position  6  is  8
>>> main()
Enter the position of the desired number in the Fibonacci sequence? 13
The Fibonacci number in position  13  is  233
>>> main()
Enter the position of the desired number in the Fibonacci sequence? 25
The Fibonacci number in position  25  is  75025
```

3. (File reading, List, Decision 15 points) Write a function wordGame() that reads the text file, Pride_and_Prejudice.txt. (Be sure to save Pride_and_Prejudice.txt in the same folder in which you are saving your program.) Include code in this function that retrieves 2 random words from the file. Ask the user to guess which of the words the author used more often in the text. If the user picks the word with the higher count, then they have guessed correctly.
To pick a random word, import random and use the random.choice () method. To determine how often a word occurs (use the count method). Use a logical condition to determine if the user guessed correctly.
Note: There is only one function in this program. To run the function, type wordGame() at the IDLE prompt.

Sample:



```
>>> wordGame()
Which word did the writer use more often "such" or "liberty"? such
you are correct
>>> wordGame()
Which word did the writer use more often "their" or "would"? would
you are correct
>>> wordGame()
Which word did the writer use more often "the" or "her"? her
you are incorrect
```