Lab Week 4 Grading Rubric and Instructions

This lab is assigned for Week 4 of COM S 127: Introduction to Programming.

This lab is due by the end of the lab period seven (7) days after the one it is assigned in. See the syllabus for details.

Lab Objective

The purpose of this lab is for students to familiarize themselves with creating functions, and assigning the function output to a variable when the function returns.

Instructions/ Deliverables

NOTE: These tasks can be completed in any order you like. See the **Grading Items** section below for the point distribution.

CITATION: Many of the exercises found here could possibly be seen as adaptations of exercises found in the online textbook “How to Think Like a Computer Scientist: Interactive Edition” By Jeffrey Elkner, Peter Wentworth, Allen B. Downey, Chris Meyers, and Dario Mitchell.

- Available: https://runestone.academy/ns/books/published/thinkcspy/index.html?mode=browsing
- Accessed: 2-2-2023
- The abbreviation 'thinkcspy' and the chapter/ section number will be used to indicate where similar exercises can be found. This citation will be placed next to the exercise title.
    - ex: [thinkcspy 2.13] indicates a similar exercise can be found in chapter 2, section 13.

Reading:

- Read Runestone chapter 6, and show the TA the notes you took in your Engineering Notebook for this chapter once you are done.
    - NOTE: You do not need to complete any of the exercises at the end of the chapter. However, it would be helpful to you in the long term if you were to do so.

`randomSum.py` [thinkcspy 5.7]

- Take input from the user in the form of three integers - `a`, `b`, and `c`.
- Write a function that takes the three integers as input.
    - This function be called `randomSum()`, and it should calculate the sum of `c` random numbers in a range between `a` through `b + 1` (non-inclusive).
        - For example, if `a = 1`, `b = 6`, and `c = 5`, then we would sum together 5 random numbers between 1 and 6 (as `b = 6` and `6 + 1 = 7`, but the function we want is non-inclusive).
        - HINT: This task will certainly involve using a 'for loop.'
- Return the sum calculated in the `randomSum()` function, and assign this output to a variable.
- Print the contents of the variable containing the return value of `randomSum()`.
- Save your code, including your name, code creation date, lab number, and brief description of what your code does, to a file called `randomSum.py`.

`sqrtIter.py` [thinkcspy 6.13]

- Review the following webpage, https://www.cuemath.com/algebra/square-root-of-2/ (accessed 2-2-2023) - particularly the section labeled "Square Root of 2 by Estimation and Approximation Method."
    - Notice the value of 'x' in the formula. This is the value we want to find the square root of.
    - Consider that we can extrapolate the information in "the iteration formula" to make a general-purpose function to calculate *any* square root - not just the square root of two (2).
    - NOTE: Be sure to cite the website above in your code.
- In your code, take two (2) integer input values from the user. One value for `x`, and the other called `iterations`.
    - `x` will store the number we want to calculate the square root of.
    - `iterations` will store how many times we want to do a 'for loop' on the formula.
- Create a function, called `sqrtIter()`, which takes both `x` and `iterations` as parameters. -This function will return the square root of `x`. - You will calculate the square root of `x` by running the "estimation and approximation method" shown on the website. - You will run this method in a 'for loop' the number of times specified by the 'iterations' variable.
    - HINT: We are calculating and returning `y`, as seen in the formula.
    - HINT: The 'initial guess' for 'y' can be stated in your code as `y = (x + 1) / 2`. This 'initial guess' is calculated from the formula `y = ((x / y_0) + y_0) / 2`, where the initial value for `y`, written here as `y_0` in the previous formula, has the value of one (1).
        - NOTE: This 'initial guess' computation should be the first thing that happens in your function - before your 'for loop.'
        - After calculating the 'initial guess,' You can then continually recalculate 'y' in your subsequent 'for loop.' You can do this using the formula `y = ((x / y) + y)/2`, as seen on the website.
            - This will work because `y` will have already been supplied with an initial value from the 'initial guess' the line above the 'for loop.'
            - If we did not have the 'initial guess,' and just tried to run the 'for loop' directly, we would have a 'variable undefined error.' This is because we would be attempting to assign something to our value for `y`, using `y` in the expression to assign to `y`, before `y` even has a value inside of it to begin with.
    - HINT: This task will certainly involve using a 'for loop.'
- Print the contents of a variable containing the return value of `sqrtIter()`. Meaning - you must assign the output of `sqrtIter()` to a variable and print it. Do not just do the following:

```
print(sqrtIter())
```

- Save your code, including your name, code creation date, lab number, and brief description of what your code does, to a file called `sqrtIter.py`.

`tridecagonTurtle.py` [thinkcspy 6.13]

- Prompt the user to enter three numerical integer values - `s`, `x`, and `y`.
- Write a function, called `tridecagonTurtle()`, which takes `s`, `x`, and `y` as parameters, and use the turtle library to draw a shape called a **regular tridecagon** at the location and at the size specified by the user.
    - The `s` value specifies the length of one of the sides of the regular tridecagon.
    - The `x` and `y` values specify the `(x, y)` coordinates of the regular tridecagon's first point (called a vertex).

    - HINT: You may need to perform an internet search to learn what this shape is, and what angle occurs between each of its segments. You will need to cite this information in a comment. Be sure to include the URL, author (if available), date the article was written, and the date you accessed the webpage.
    - HINT: This task will certainly involve using a 'for loop.'
- Save your code, including your name, code creation date, lab number, and brief description of what your code does, to a file called `tridecagonTurtle.py`.

Files Provided

None

Example Script

```
# Matthew Holman              2-3-2023
# Lab Week 4 - This code calculates the value of two numbers the user inputs.

def add(a, b):
        return a + b

a = int(input("Enter value 'a': "))
b = int(input("Enter value 'b': "))

c = add(a, b)

print("The result of {0} + {1} = {2}".format(a, b, c))
```

Example Output

```
Enter value 'a': 2
Enter value 'b': 3
The result of 2 + 3 = 5
```

Grading Items

- (**Attendance**) Did the student attend the lab meeting, or make arrangements to attend virtually via WebEx?: ______ / 1
- (**Reading**) Has the student read chapter 6 of the Runestone textbook and shown their notes in their Engineering Notebook to the TA?: ______ / 2
- (**randomSum.py**) Has the student completed the task above, and saved their work to a file called `randomSum.py`?: ______ / 2
- (**sqrtIter.py**) Has the student completed the task above, and saved their work to a file called `sqrtIter.py`?: ______ / 3
- (**tridecagonTurtle.py**) Has the student completed the task above, and saved their work to a file called `tridecagonTurtle.py`?: ______ / 2

TOTAL ______ / 10