CS 100
Homework 08

Due Date:

Do all of the items below and **submit** one ZIP file containing all the solutions via Canvas. If you run into a problem, post to Canvas describing where you ran into trouble or email your instructor or classroom assistant, or ask your question during recitation hours. If you know the answer to someone's question on Canvas, post a response. You get course credit for asking and answering questions in Canvas.

- Read Chapter 7 (Iteration) in the textbook.

- Read the Python tutorial section 4.4 (break and continue Statements, and else Clauses on Loops). The Python tutorial can be accessed through the documentation installed with IDLE:
  Help → Python Docs → Tutorial → 4. More Control Flow Tools
  If you are using an alternate IDE, visit:
  https://docs.python.org/3/tutorial/controlflow.html#more-control-flow-tools
  to browse the tutorial online.

- In the Python editor IDLE, create and save a Python file for each problem given. Before submitting your solutions via Canvas, ZIP the Python files and name the archive, if your name is Harry Houdini, for example, *HW8_HarryHoudini.zip*. Each Python file must begin with a comment containing your name, class and section, the posting date and number of the homework assignment.

Problem 1

This problem provides practice using a `while True` loop.

Write a function named `twoWords` that gets and returns two words from a user. The first word is of a specified length, and the second word begins with a specified letter.

The function `twoWords` takes two parameters:

1. an integer, `length`, that is the length of the first word and
2. a character, `firstLetter`, that is the first letter of the second word. The second word may begin with either an upper or lower case instance of `firstLetter`.

The function `twoWords` should return the two words in a list. Use a `while True` loop and a `break` statement in the implementation of `twoWords`.

The following is an example of the execution of `twoWords`:

```
print(twoWords(4, 'B'))
Enter a 4-letter word please: two
Enter a 4-letter word please: one
Enter a 4-letter word please: four
Enter a word beginning with B please: apple
Enter a word beginning with B please: pear
Enter a word beginning with B please: banana
['four', 'banana']
```

Problem 2

Write a function named `twoWordsV2` that has the same specification as **Problem 1**, but implement it using **`while`** and not using **`break`**. (Hint: provide a different boolean condition for **`while`**.)

Since only the implementation has changed, and not the specification, for a given input the output should be identical to the output in **Problem 1**.

Problem 3

Write a function named `enterNewPassword`. This function takes no parameters. It prompts the user to enter a password until the entered password has 8-15 characters, including at least one digit. Tell the user whenever a password fails one or both of these tests.

Problem 4

Implement the GuessNumber game. In this game, the computer

- Think of a random number in the range 0-50. (Hint: use the random module.)
- Repeatedly prompt the user to guess the mystery number.
- If the guess is correct, congratulate the user for winning. If the guess is incorrect, let the user know if the guess is too high or too low.
- After 5 incorrect guesses, tell the user the right answer.

The following is an example of correct input and output.

```
I'm thinking of a number in the range 0-50. You have five tries to guess it.
Guess 1? 32
32 is too high
Guess 2? 18
18 is too low
Guess 3? 24
You are right! I was thinking of 24!
```