CS 100
Homework 10

Do all of the items below and **submit** one .py 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 11 (Dictionaries) in the textbook.
- Read the Python tutorial section 5.5 (Dictionaries). The Python tutorial can be accessed through the documentation installed with IDLE:
  Help → Python Docs → Tutorial → 5.5. Dictionaries
  If you are using an alternate IDE, visit:
  https://docs.python.org/3/tutorial/datastructures.html#dictionaries
  to browse the tutorial online.
- **Write and test**. For each problem, write code that satisfies the problem specification. Test your code by designing at least three inputs for each problem that you believe are likely to reveal any bug and run your code on those inputs. **Include** the test inputs in the `.py` file.

Problem 1

Write a function named *`initialLetterCount`* that takes one parameter, *`wordList`* — a list of words. Create and return a dictionary in which each initial letter of a word in *`wordList`* is a key and the corresponding value is the number of words in *`wordList`* that begin with that letter. The keys in the dictionary should be case-sensitive, which means `'a'` and `'A'` are two different keys.

For example, the following is correct output:

```
horton = ['I', 'say', 'what', 'I', 'mean', 'and', 'I', 'mean', 'what', 'I', 'say']
print(initialLetterCount(horton))
{'I': 4, 's': 2, 'w': 2, 'm': 2, 'a': 1}
```

Problem 2

Write a function named *`initialLetters`* that takes one parameter, *`wordList`* – a list of words. Create and return a dictionary in which each initial letter of a word in *`wordList`* is a key and the corresponding value is a list of the words in *`wordList`* that begin with that letter. There should be no duplicate words in any value in the dictionary.

For example, the following is correct output:

```
print(initialLetters(horton))
{'I': ['I'], 's': ['say'], 'w': ['what'], 'm': ['mean'], 'a': ['and']}
```

Problem 3

Write a function named *shareALetter* that takes one parameter, *wordList* – a list of words. Create and return a dictionary in which each word in *wordList* is a key and the corresponding value is a list of all the words in *wordList* that share at least one letter with that word. There should be no duplicate words in any value in the dictionary.

For example, the following is correct output:

```
print(shareALetter(horton))
{'I': ['I'], 'say': ['say', 'what', 'mean', 'and'], 'what': ['say', 'what', 'mean',
'and'], 'mean': ['say', 'what', 'mean', 'and'], 'and': ['say', 'what', 'mean', 'and']}
```