

## CS 100

### Homework 09

#### 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 9 (Case study: word play), and sections 14.1, 14.2, 14.3, and 14.4 from Chapter 14 (Files) in the textbook.
- 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, *HW9\_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

Write a function *file\_copy* that takes two string parameters (*in\_file* and *out\_file*) and copies the content of *in\_file* into *out\_file*. Assume that *in\_file* exists before *file\_copy* is called. For example, the following would be correct input and output:

```
>>> file_copy('created_equal.txt', 'copy.txt')
>>> copy_f = open('copy.txt')
>>> copy_f.read()
'We hold these truths to be self-evident,\nthat all men are created equal\n'
```

#### Problem 2

Write a function named *file\_stats* that takes one string parameter (*in\_file*) that is the name of an existing text file. The function *file\_stats* should calculate three statistics about *in\_file*: the number of lines it contains, the number of words and the number of characters, and print the three statistics on separate lines. For example, the following would be correct input and output:

```
>>> file_stats('created_equal.txt')
lines 2
words 13
characters 72
```

*Note: The number of characters may vary slightly between operating systems. Similarly, the number of lines may vary by 1 line, depending on the method used to calculate it.*

### Problem 3

Write a function named *repeat\_words* that takes two string parameters:

1. *in\_file*: the name of an input file that exists before *repeat\_words* is called
2. *out\_file*: the name of an output file that *repeat\_words* creates

Assume that the input file is in the current working directory and write the output file to that directory.

For each line of the input file, the function *repeat\_words* should write to the output file all of the words that appear more than once on that line. Each word should be lower cased and stripped of leading and trailing punctuation. Each repeated word on a line should be written to the corresponding line of the output file only once, regardless of the number of times the word is repeated.

For example, if the following is the content of the file *catInTheHat.txt*:

```
Too wet to go out and too cold to play ball.  
So we sat in the house.  
We did nothing at all.  
So all we could do was to Sit! Sit! Sit! Sit!
```

The following function call:

```
inF = 'catInTheHat.txt'  
outF = 'catRepWords.txt'  
repeat_words(inF, outF)
```

should create the file *catRepWords.txt* with the content:

```
too to
```

```
sit
```

*Hint: Be sure to test your solution with input in which some repeated words on a line are a mixture of upper and lower case, and in which repeated words sometimes are preceded or followed by punctuation.*