

## LAB 05

### SUBMISSION INSTRUCTIONS

Submit 2 python files using the naming convention below (replace **JaneDoe** with your first and last name respectively):

- JaneDoe5\_1.py - For question 1.
- JaneDoe5\_2.py - For question 2.

### QUESTIONS

1. Write a script that will:
  - a. Create an output file named **test.txt** using the **w+** mode (**w+** means we will first write to the file then read from it). The file should be created in the same folder as your script.
  - b. Store integers from the user to the file (Use the sample output to determine how exceptions should be handled).
  - c. Once the user enters **done**, go to the beginning of the file (this can be done by setting the read position back to 0 using **file.seek(0)** where **file** would be the file object name).
  - d. Read the file data to determine the smallest number, largest number, total count of numbers, and the sum of all numbers stored in the file. Display these values to the console with the help of the **print** function.

Please note:

- Write your program as a script (i.e. include the if `__name__ == '__main__':` block).
- Use the file names provided in the instructions.
- Don't forget to close any files you open.

### SAMPLE OUTPUT

Opened file: test.txt

Enter a number: 4

Enter a number: 5

Enter a number: data

Invalid input

Enter a number: 7

Enter a number: done

Data stored to test.txt

Minimum = 4

Maximum = 7

Count = 3

Sum = 16

2. Complete the steps below:
  - a. Copy and paste the **files** folder to your python project.
  - b. Ask the user for the input file name (use exceptions to deal with a missing file).

- c. Once the right file name is provided, look for lines that begin with **From:** and extract the email address.
- d. Keep track of how many times an email is sent from a particular address with the help of a dictionary.
- e. Send your dictionary results to an output file called **output.txt** which must be stored in the **files** folder.
- f. Use the **os.path** module (not exception handling) to check if the output file already exists. If it does, ask the user if they want to overwrite it or if they would prefer creating a different output file (refer to sample output for the logic). Regarding the **os** module, the code below is enough to get you started.

```
import os.path
if os.path.isfile("Presidents.txt"):
    print("Presidents.txt exists")

Here isfile("Presidents.txt") returns True if the file Presidents.txt exists in the
current directory.
```

- g. Your output file should have data identical to the **sample.txt** file.

Please note:

- Write your program as a script (i.e. include the `if __name__ == '__main__':` block).
- Use the relative file names for input and configure the **open** function to look for the files inside of the **files** folder.
- Use the input and output file names provided in the instructions.
- File names are case sensitive.
- String input should be stripped.
- Don't forget to close any files you open.
- Feel free to use functions to organize your code.

### **SAMPLE OUTPUT 1**

Input file name: **test.txt**

File does not exist!

Input file name: **input**

File does not exist!

Input file name: **input.txt**

Output file name: **sample.txt**

Overwrite existing file (y/n): **x**

Enter (y/n): **a**

Enter (y/n): **N**

New output file name: **sample.txt**

Overwrite existing file (y/n): **n**

New output file name: **output.txt**

Data stored!

**NOTE:** The assumption is that you didn't have a file named output.txt in your files folder.

### **SAMPLE OUTPUT 2**

Input file name: **input.txt**

Output file name: **output.txt**

Overwrite existing file (y/n): **y**

**Data stored!**

NOTE: The assumption is that you had a file named output.txt in your files folder.

### **SAMPLE OUTPUT 3**

Input file name: **input.txt**

Output file name: **output.txt**

**Data stored!**

NOTE: The assumption is that you didn't have output.txt in your files folder so this is the first time it's being created.