**05: Lists & Dictionaries**

**Introduction**

The culminating project for this week was a continuation of the previous assignment. It still involved working with reading and writing collections to and from files but in this case involved working with dictionaries and pre-written code; the latter working as an introduction to the common scenario of interpreting code developed by other programmers.

**Planning and Writing**

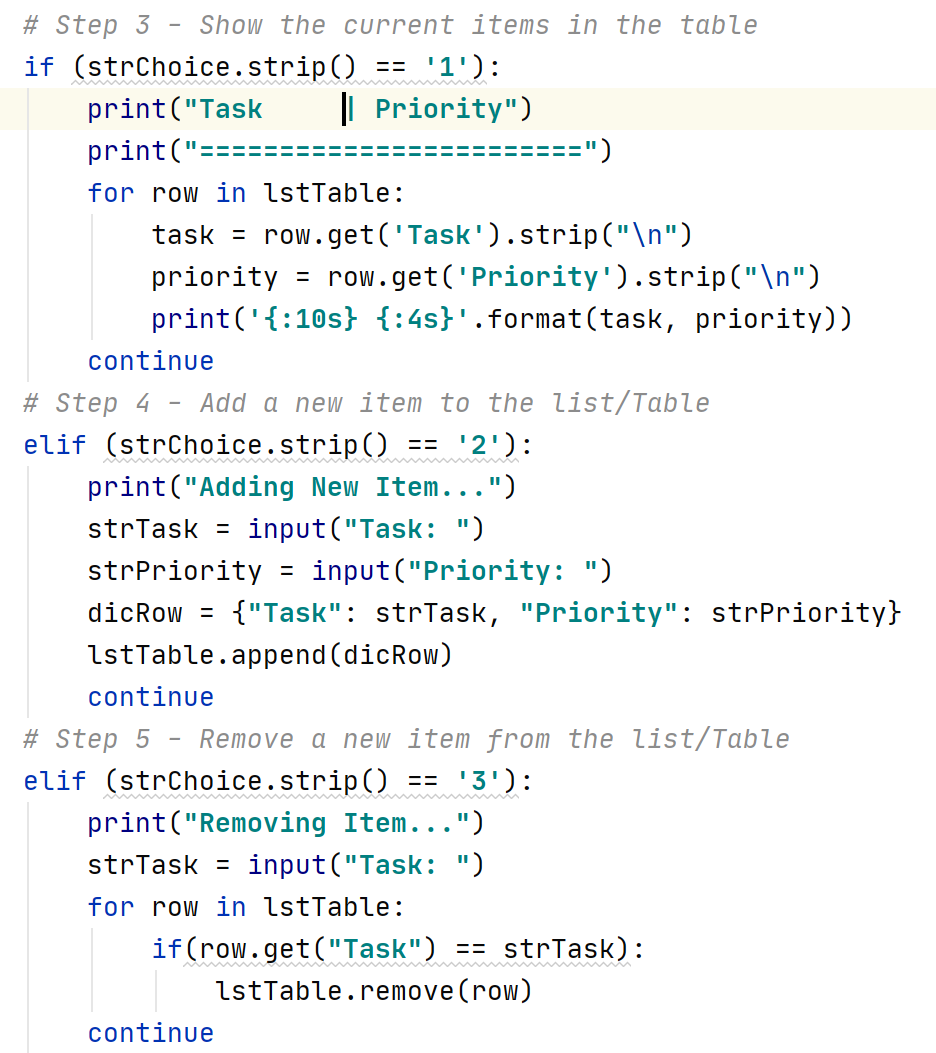
Like the previous two assignments the objective was to create a program that could take user input and save that input into a file. Similar to last week the user needed the ability to write and save multiple entries into a file but to also read a preexisting file. In this case the file was to be interpreted as a list of dictionaries acting as a table.

The table in question had each dictionary representing an entry/row with two column/categories: A Task (The Key) & it’s numerical Priority (The Value). The starter code already instantiated the majority of the variables and a navigable menu so what remained was implementing the various menu options and file reading & writing.

**Menu Implementation**

Broken down, the menu options consisted of printing, adding, removing, and saving to file a list of entries.

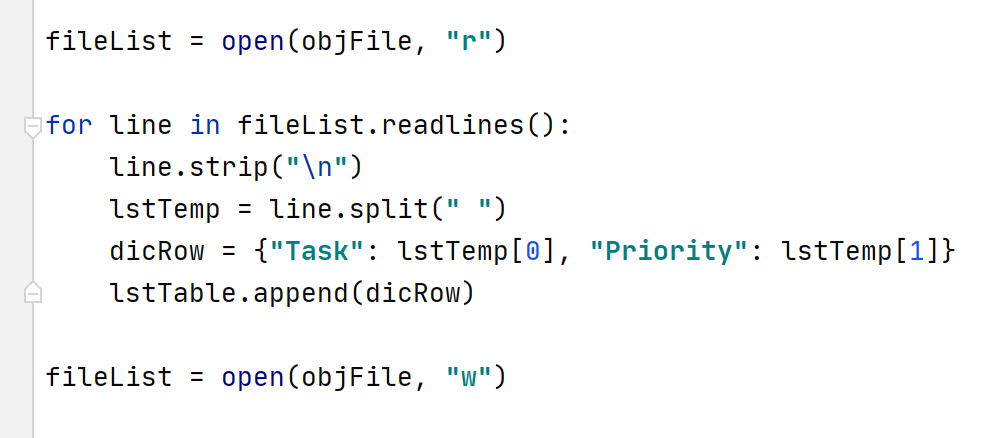
Adding required two prompts for input: a ‘Task’ as the key and a ‘Priority’ for the value; both were put into a new dictionary that made the next entry to be appended into the tabular list. Removing prompted for what task to look for and the script iterated through the tabular list and removed the dictionary with the given key. Finally printing the table solely involved iterating through the list of dictionaries and printing out the contents. (Figure 1)



*Figure 1: Printing Table, Adding Entry, Removing Entry*

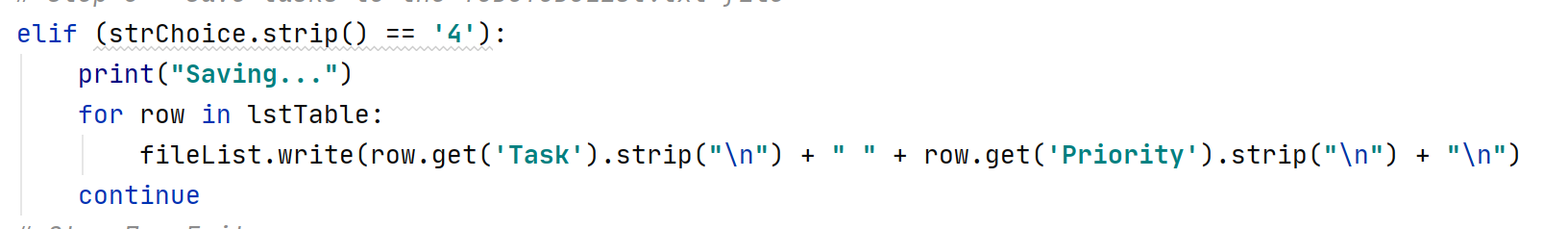
**File Reading and Writing**

Reading and writing to file required the same process. Each line of the text file represented one entry (a Key-Value pair). To read the file was iterated through as a list using the readlines() function with every line converted to a dictionary added to the tabular list.



*Figure 2: Reading from file*

To write was the opposite, with the tabular list iterated through with each dictionary turned into a string to be written as a new line for the file (Figure 3).

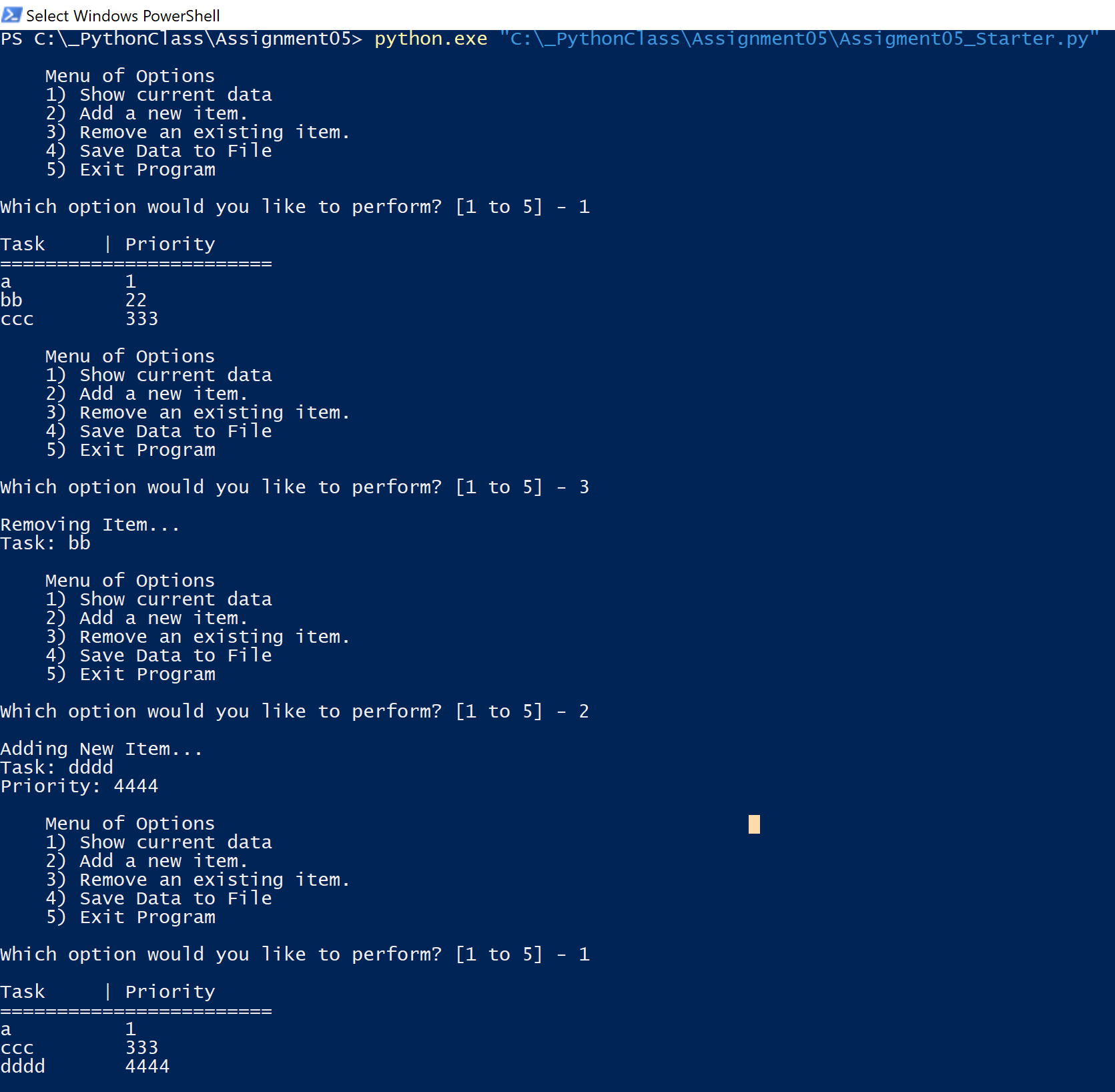
*Figure 3: Writing to file*

**Testing**

The script was tested in Windows Powershell using a text file containing the following dictionaries:

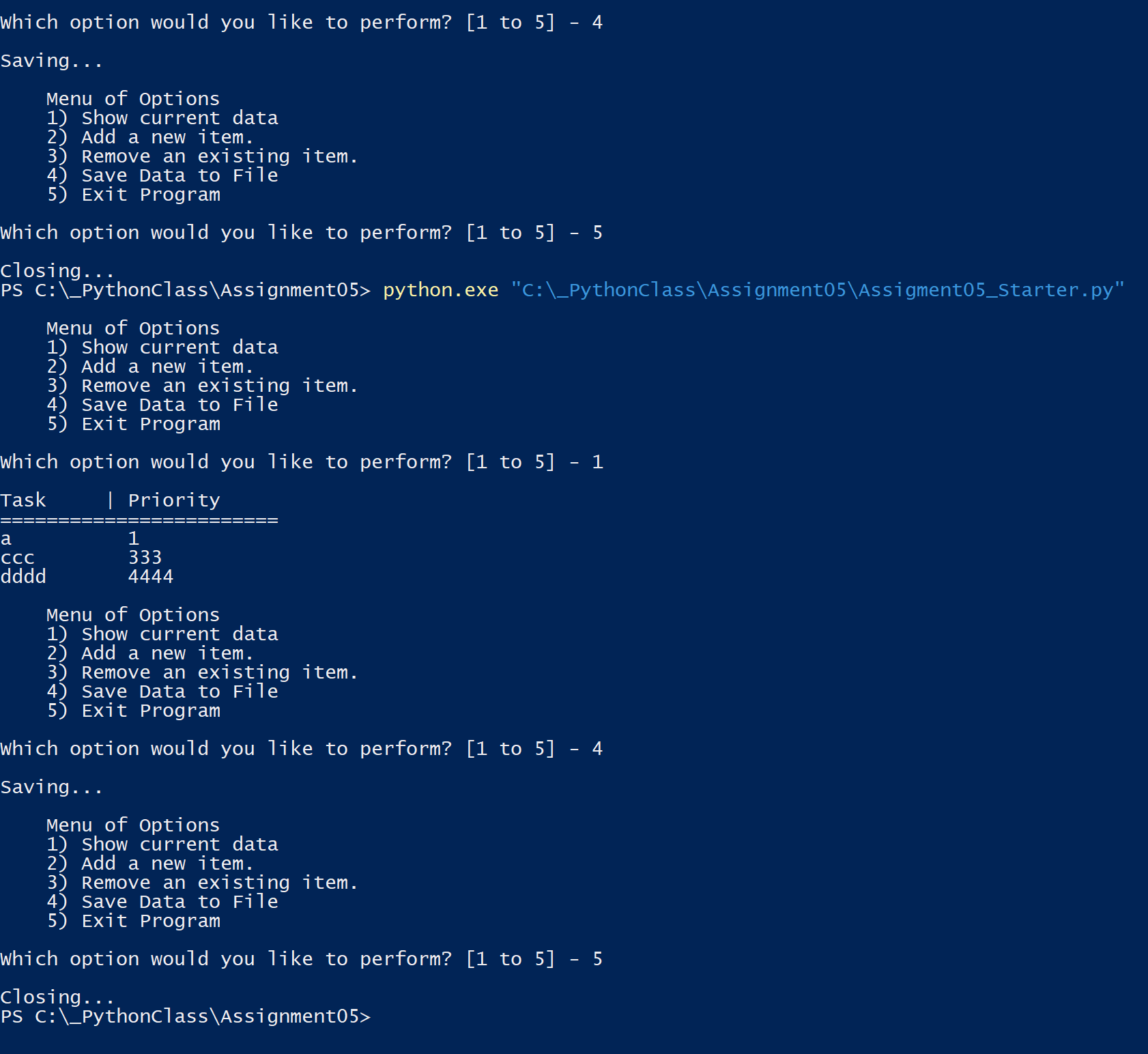
|  |  |
| --- | --- |
| **Task** | **Priority** |
| a | 1 |
| bb | 22 |
| ccc | 333 |

The table was first printed then task ‘bb’ was removed and a task ‘dddd’ was added (Figure 4).



*Figure 4: Testing through Shell (Print, Add, & Remove)*

The table was saved to file and when the script was ran again the file was read and the changes were shown to be persistent (Figure 5).

*Figure 5: Testing through Shell 2 (Read & Write)*

**Summary**

Like the previous assignment the challenge came from formatting to and form files. Manipulating collections like lists and dictionaries alone are easier due to them being sequences of values. How to convert data to usable and storable data required trial and error.