List and Dictionaries

**Introduction**

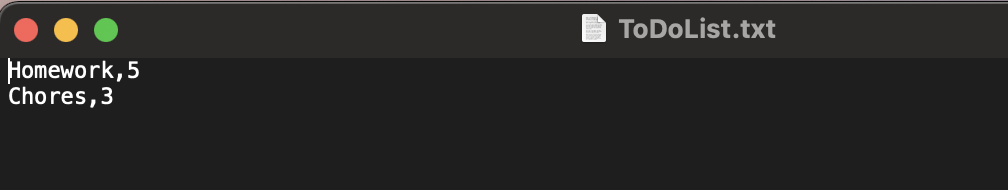
In this module, we mainly focused on the aspect of manipulating data within and into a text file. The two methods of doing so were to use lists and dictionaries to load and store data into a file. While lists were something we have partially covered in previous module, dictionaries are a new topic that we touched upon. There are many similarities between the sequences, but there are also key differences that set it apart. The other subject we touched upon in module 5 was how to improve your script through various ways. In the end, our understanding of the material was tested via the task of taking a script template and completing it to allow a user to see a menu with multiple options.

# **Dictionaries and Separation of Concerns**

Dictionaries are a type of sequence, just as are strings, tuples, and lists. The main difference is that dictionaries access their elements using character subscripts rather than numeric subscripts, as well as having the braces operator indicate its usage. Moving away from all the operators we have been learning recently, we shift our attention to a more logistical aspect of coding. With the additional complexity that comes with having a larger script, it becomes important to be able to organize properly. One method of doing is called the separation of concerns and it is mainly concerned with, “separating a computer program into distinct sections, so that each section addresses a separate concern” (<https://en.wikipedia.org/wiki/Separation_of_concerns>, 2019) (External Site). The sections we are concerned with include data, processing, and presentation.

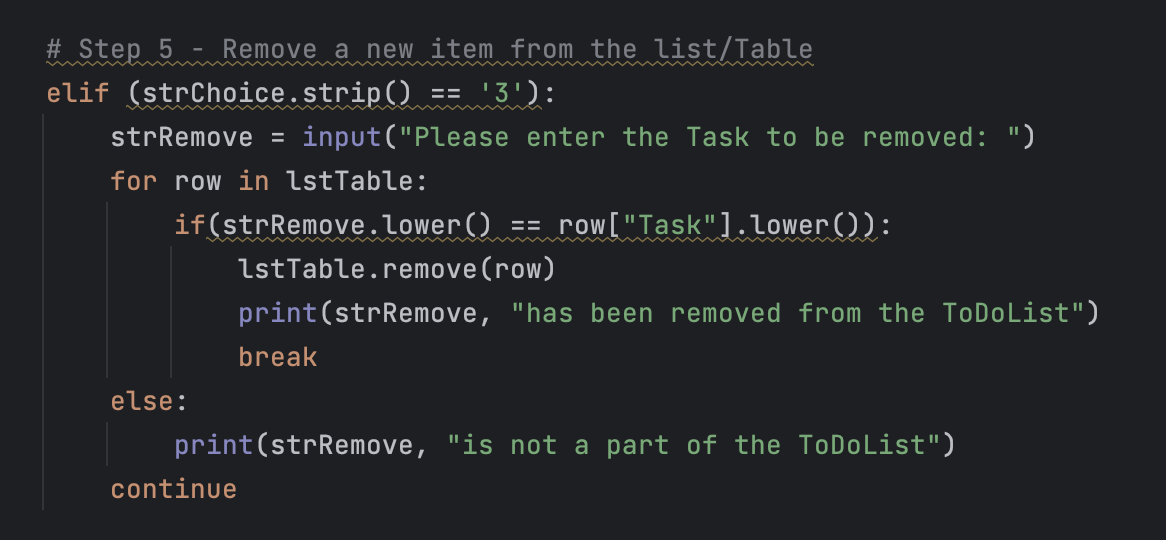
# **Module 5 Script Assignment**

This week’s assignment was unique for multiple reasons. First, it was only assignment so far where a template script was provided. Having to read through a foreign script and decipher its meaning was an interesting experience and I believe it is something that we need to get used to when coding in a professional environment. I began the assignment by creating a ToDoList file that the script would use to read data from, as seen in Figure 1.



*Figure 1: Initial ToDoList.txt before running script.*

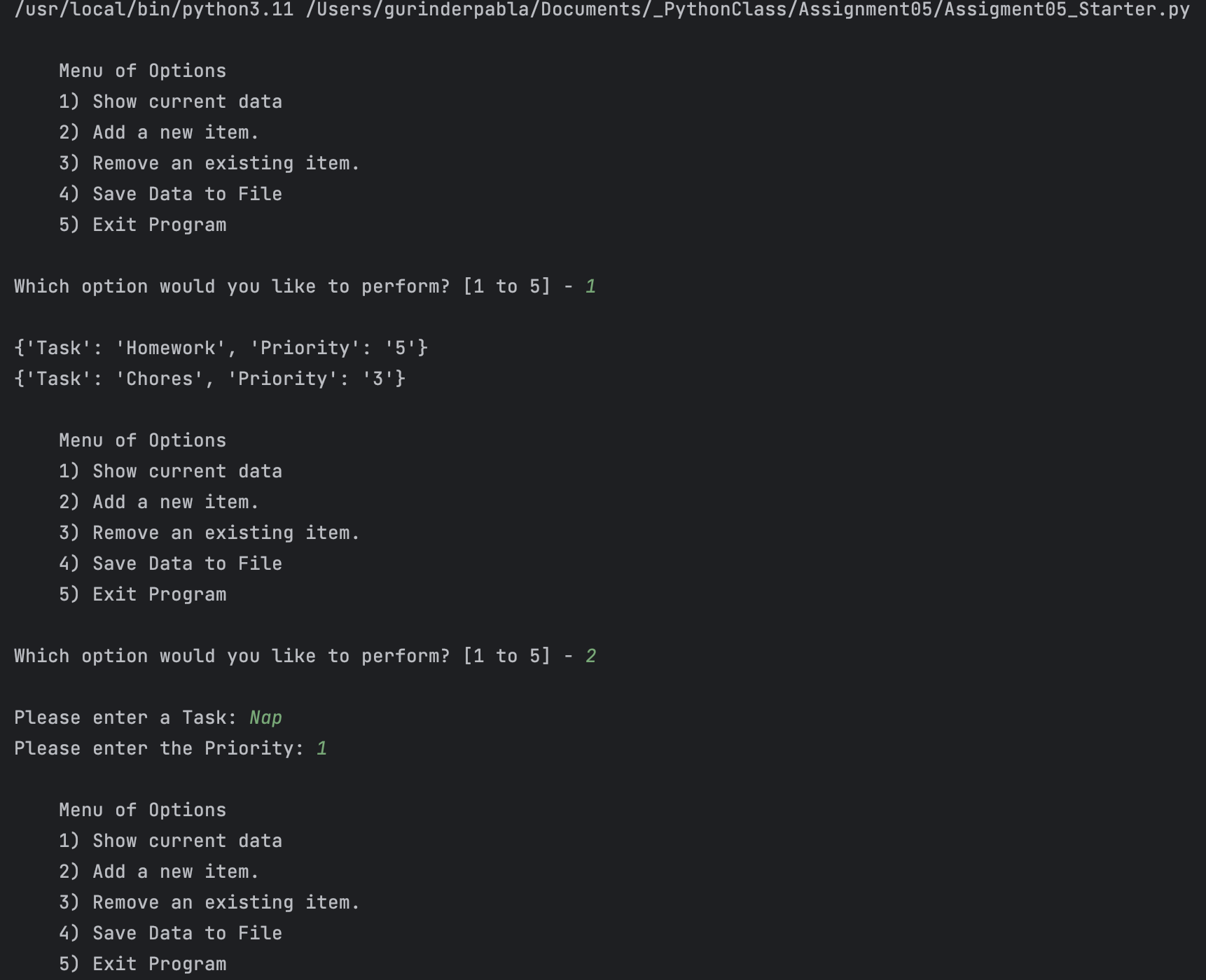
The next step was to take that data and have it set as a list of dictionary rows. From there, we needed to have the script successfully complete the options set in menu. There were many similarities here from last week, but the main challenge was removing an item from this list. This required familiarity with how lists and dictionaries function and how to be able to use the keys to call the value to be removed, as shown in Figure 2. Figures 3-9 shows the complete run of the script on both PyCharm and the Terminal.



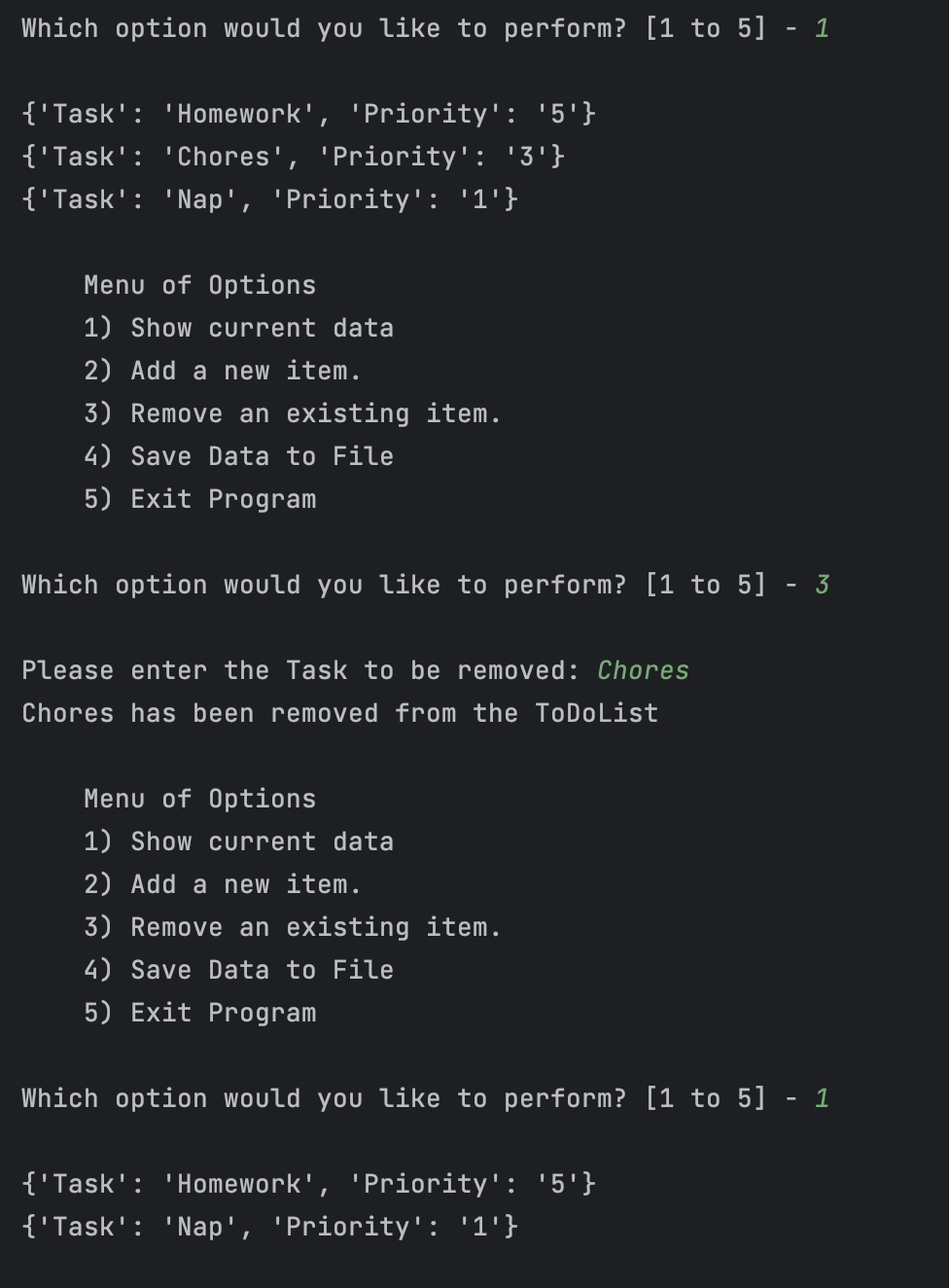
*Figure 2: Coding for removing an item from a list*

# **Summary**

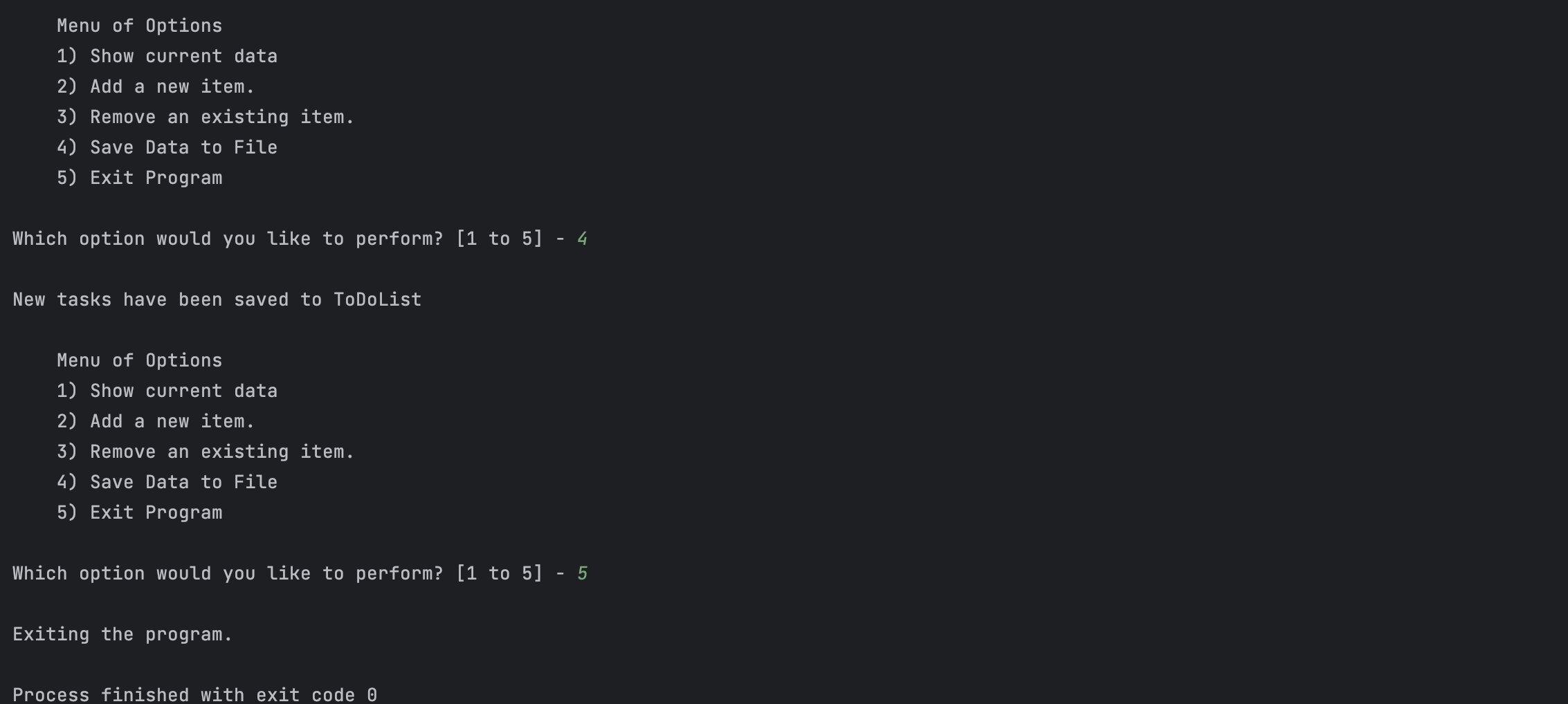
This week’s module allowed us to learn an additional sequence in the form of a dictionary, with the biggest difference. We also took an initial look at one method of organizing our scripts to ensure efficiency of code and ease of understanding from others who will need to read the script. Overall, this week was definitely challenging, and it was just not limited to the class material. Work has been requiring a lot of overtime for the past two weeks and it has impacted my usual routine for my work-school balance. I will strive to manage my time better in the future, but for now, I appreciate the extension for this week’s assignment.



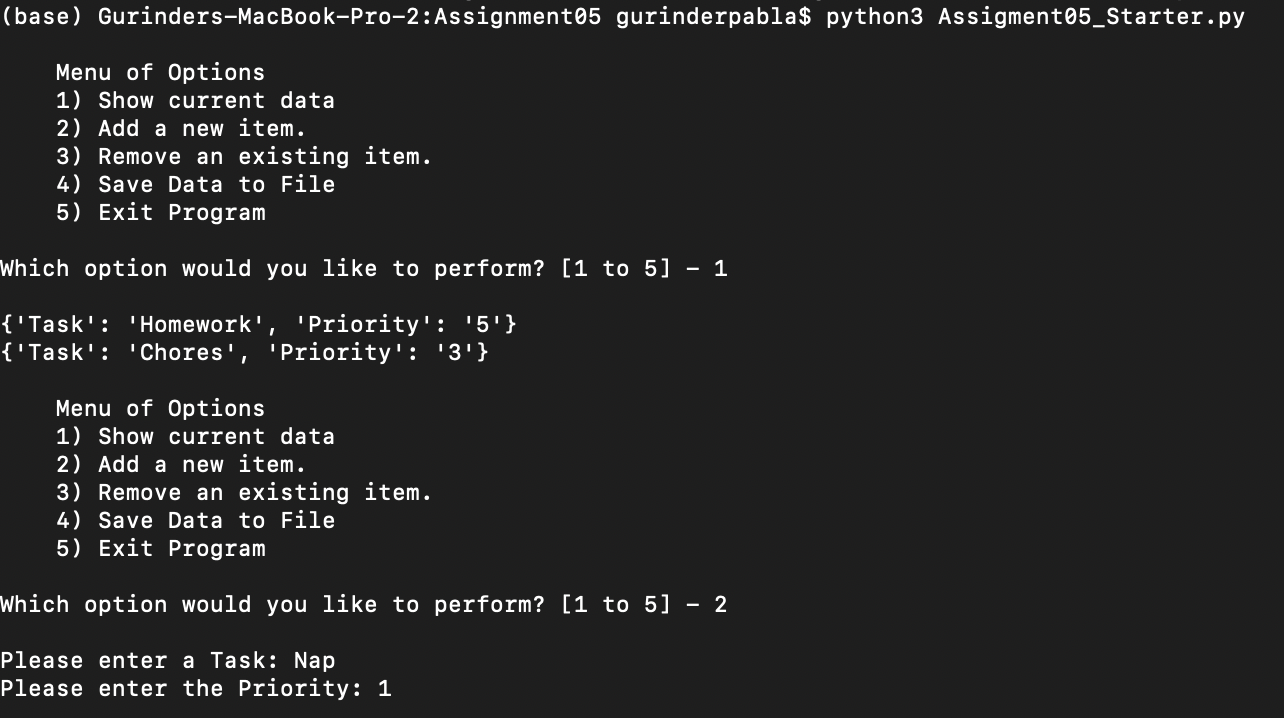
*Figure 3: Script on PyCharm*



*Figure 4: Script on PyCharm*



*Figure 5: Script on PyCharm*



*Figure 6: Script on Terminal*

A screenshot of a computer program

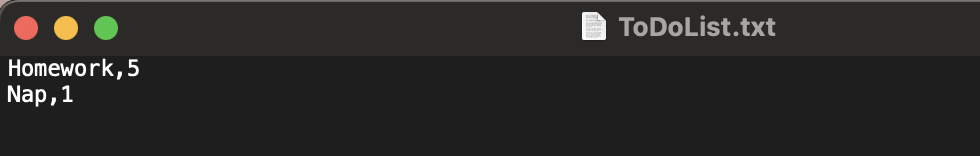
Description automatically generated

*Figure 7: Script on Terminal*

A screenshot of a computer program

Description automatically generated

*Figure 8: Script on Terminal*



*Figure 9: End result of script*