Irina Zubova

Feb 14, 2021

Foundations of Programming: Python

Assignment 05

https://github.com/i-zuzu/IntroToProg-Python

Modifying a Script that Manages "ToDo list"

Introduction

In this document I'll describe in details how I worked on Assignment 05. The goal was to get the script that would manage "ToDo list" text file and performed the following actions:

1) Show current data
2) Add a new item.
3) Remove an existing item.
4) Save Data to File
5) Exit Program

The starter script was provided. I created the "ToDo list" file with two columns of data, "Task" and "Priority." Both the starter Python file and the "ToDo list" were saved in **Assignment 05 folder.**

This assignment will help me to practice in:

- Reading and understanding the code written by other person
- Creating dictionaries in Python
- Accessing specific data in the dictionaries
- Using Lists and Dictionaries to store collections of data in memory and text files
- Opening and Modifying data in the text files
- Representing data from the text file in virtual table using lists and dictionaries
- Display data from the lists/dictionaries to the user in the desired format

Drafting the Script

1. Checking the starter script

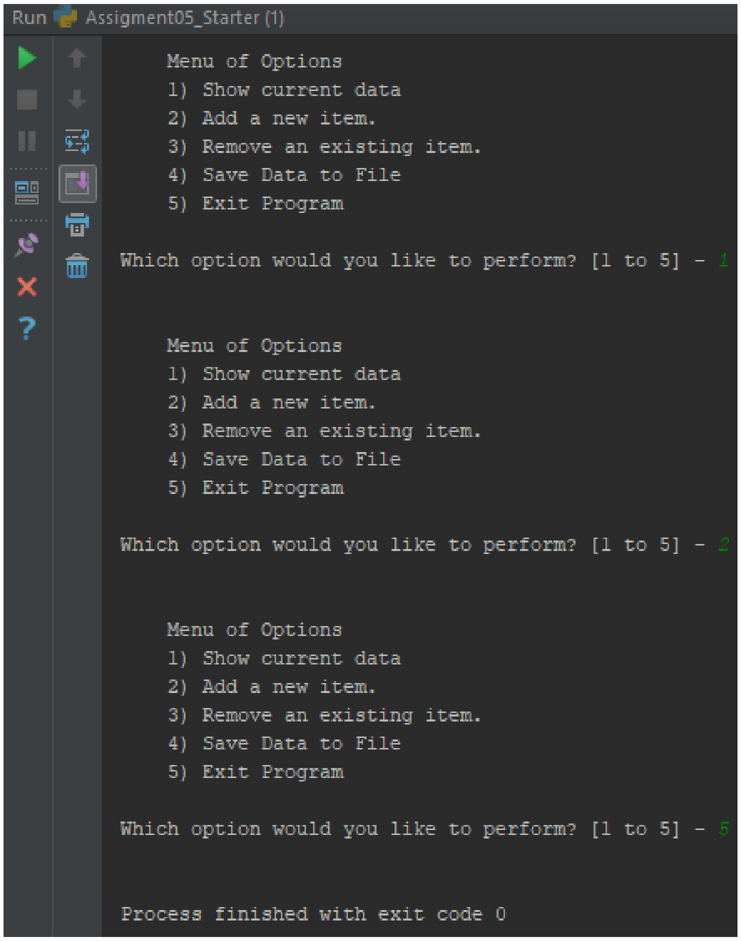
I opened provided Assignment05_Starter script in PyCharm and checked that it works. The script had several sections and the pseudo code (what you need to do to write the working code) with 7 steps:

- "Script Header" to describe what the script does and keep the revision control
- "Data" to declare variables and constants that will be used in the script
- "Processing" to load the data
- "Input/Output" to interact with the user

Below is the starter script:

```
# ------------------------------------------------------------------------
----- #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#              When the program starts, load each "row" of data
#              in "ToDoToDoList.txt" into a python Dictionary.
#              Add the each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# Irina Zubova,2-14-2020,Added code to complete assignment 5
# ----------------------------------------------------------------------- #

# -- Data -- #
# declare variables and constants
objFile = "ToDoList.txt"   # An object that represents a file
strData = ""  # A row of text data from the file
dicRow = {}    # A row of data separated into elements of a dictionary
{Task,Priority}
lstTable = []  # A list that acts as a 'table' of rows
strMenu = ""   # A menu of user options
strChoice = "" # A Capture the user option selection


# -- Processing -- #
# Step 1 - When the program starts, load the any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows
(like Lab 5-2)
# TODO: Add Code Here

# -- Input/Output -- #
# Step 2 - Display a menu of choices to the user
while (True):
    print("""
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program
    """)
    strChoice = str(input("Which option would you like to perform? [1 to 5] -
"))
    print()  # adding a new line for looks
    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):
        # TODO: Add Code Here
        continue
    # Step 4 - Add a new item to the list/Table
    elif (strChoice.strip() == '2'):
        # TODO: Add Code Here
        continue
    # Step 5 - Remove a new item from the list/Table
    elif (strChoice.strip() == '3'):
        # TODO: Add Code Here
        continue
    # Step 6 - Save tasks to the ToDoToDoList.txt file
    elif (strChoice.strip() == '4'):
        # TODO: Add Code Here
        continue
    # Step 7 - Exit program
    elif (strChoice.strip() == '5'):
        # TODO: Add Code Here
        break  # and Exit the program
```

Figure 1 shows the outcome of the starter script after running in PyCharm.

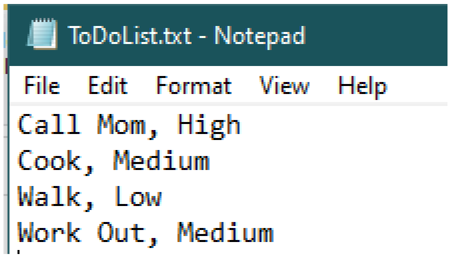


```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1


    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2


    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 5


Process finished with exit code 0
```

Figure 1. Screenshot of the Starter Script running in PyCharm

The program displayed the menu of options and offered to choose the option (from 1 to 5). Placeholder for conditional statements `if` and `elif` were created for each menu options. There was `while` loop in the script and the program kept showing the menu and asking user to provide the input until you enter 5 and that caused the loop to break and close the program.

Figure 2 shows the ToDo list text file I created. I will need to write the code to manage and make some changes in this file.



```
Call Mom, High
Cook, Medium
Walk, Low
Work Out, Medium
```

Figure 2. Screenshot of the ToDo List with Tasks and Priorities

I revised the revision control in the script header and moved to Processing section to add the code.

2. Processing section

Here's the pseudo code that helped me to understand what had to be done:

```
# -- Processing -- #
# Step 1 - When the program starts, load the any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows
(like Lab 5)
```

I needed a code that would open the ToDoList file (`objFile`) and read it. Then the code tells the program to load each row of data to a dictionary object `dicRow`. And these rows added to a Python List object to create a table of data `lstTable`.

Here's the code for this action:

```
objFile = open("ToDoList.txt", "r")
for strData in objFile:
   lstRow = strData.split(",")
   dicRow = {"Task":lstRow[0], "Priority":lstRow[1].strip()}
   lstTable.append(dicRow)
objFile.close()
```

`objFile`, `dicRow` and `dicRow` were declared in the Data section of the script. I added to this section `lstRow` variable to represent data in each row in the format of list:

```
lstRow = []   # A list of data in the row
```

The first line uses function `open()` to open the file in reading condition ("r"). I only provided the name of the file and the program uses the same location as the Starter script: `objFile = open("ToDoList.txt", "r")`

Since the text file had several rows `for` loop tells the program to go through each row and

1) split each row to lists and separate them with comma: `lstRow = strData.split(",")`

2) add data from each row to the `dicRow` dictionary. Each dictionary has 2 items or 2 pairs. "Task" key is for the first object from the list (with index 0) and "Priority" key is for the second one with index 1.

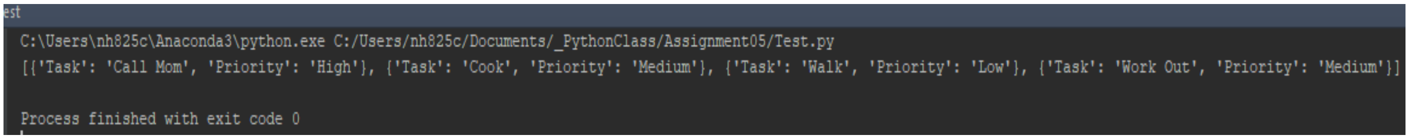
   `dicRow = {"Task":lstRow[0], "Priority":lstRow[1].strip()}`
   `strip()` method helps to delete the extra character that program creates automatically.

So if I take the first row of my To Do list file (Call Mom, High) as example the dictionary for this row will be: `dicRow = {"Task":’Call Mom’, "Priority":’High’}`

3) add dictionaries to the `lstTable` list. I used `append ()` function for this: `lstTable.append(dicRow)`

Then I just close the file since I have the data loaded to the computer memory and don’t need the file at that moment: `objFile.close()`

To check that my code works I used the print function to print the created list: `print(lstTable)`
Figure 3 below shows the print screen of the content of created `lstTable` list with dictionaries for each row of the table.



```
C:\Users\nh825c\Anaconda3\python.exe C:/Users/nh825c/Documents/_PythonClass/Assignment05/Test.py
[{'Task': 'Call Mom', 'Priority': 'High'}, {'Task': 'Cook', 'Priority': 'Medium'}, {'Task': 'Walk', 'Priority': 'Low'}, {'Task': 'Work Out', 'Priority': 'Medium'}]

Process finished with exit code 0
```

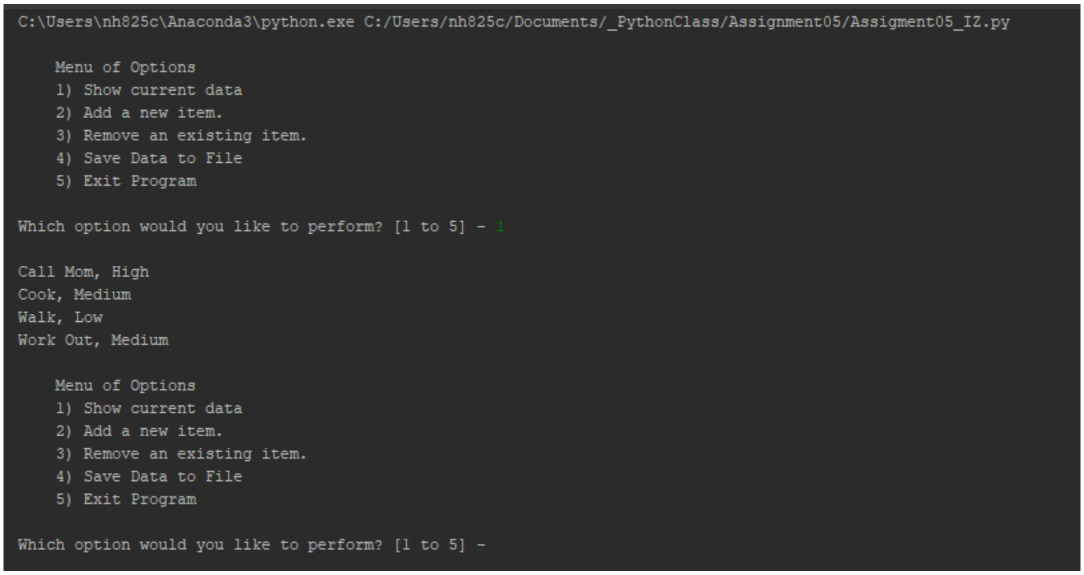
Figure 3. Screenshot of the `lstTable` list

3. <u>Show the current items in the table (Input/Output section)</u>

I now have the code to open the file and process its data and the code to display the Menu of options to the user and get input from the user. The next step is to write the code to perform Menu Option 1 which is ‘Show current data’.
Current data from the table is in the list called `lstTable`. And the code below tells the program if the user chooses Option 1 to go through each row of the table (each row is a dictionary) and display values (not the keys) in the dictionaries separated with comma:

```
if (strChoice.strip() == '1'):
    # TODO: Add Code Here
    for dicRow in lstTable:
        print(dicRow["Task"] + ", " + dicRow["Priority"].strip())
    continue
```

I checked the code running it in the PyCharm and the next figure shows the outcome – the content of the `lstTable` displayed to user in the nice template.



```
C:\Users\nh825c\Anaconda3\python.exe C:/Users/nh825c/Documents/_PythonClass/Assignment05/Assigment05_IZ.py

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

Call Mom, High
Cook, Medium
Walk, Low
Work Out, Medium

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] -
```

Figure 4. Screenshot of the outcome for the code to show current data

4. Add a new item to the list/Table (Input/Output section)

And this is option 2 from the Menu (Add a new item).
I created 2 new variables to get inputs from the user for the new item (or the Task) and Priority.

```
strTask = input("Enter the task: ")
strPrior = input("Enter Priority(Low, Medium, High): " ).title()
```

I also wanted to restrict the user to choose only from Low, Medium or High priorities so I put these 3 options in a tuple and assigned to a new constant.

```
tplPRIOR = ('Low', 'Medium', 'High')
```

Now I added while loop that continues asking the user to choose Low, Medium or High priority if he/she enters something different. Again, `title()` method helps to avoid errors related to case sensitivity and keeps the data in one format.

```
while strPrior not in tplPRIOR:
    print("Please choose Low, Medium or High")
    strPrior = input("Enter Priority(Low, Medium, High): " ).title()
```

Now the Task and Priority need to be added to dictionary that represents the row of the table:

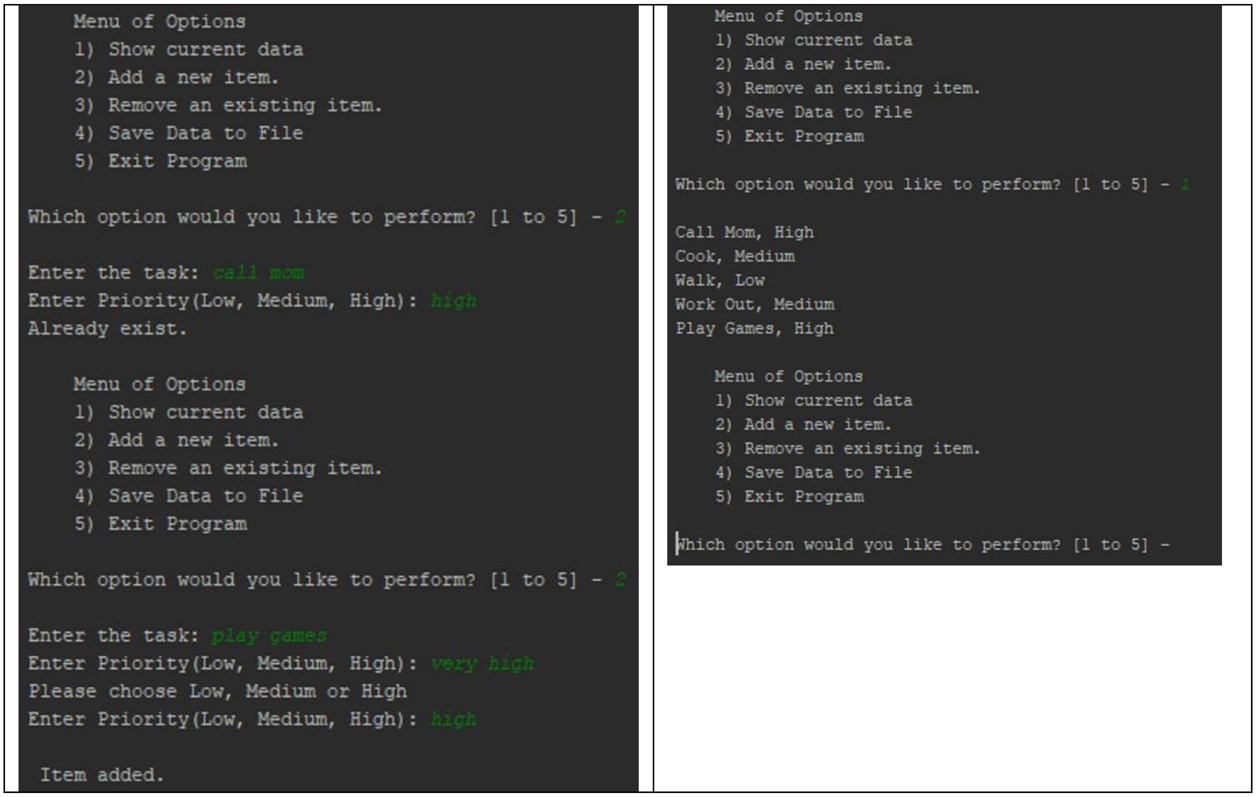
```
dicRow = {"Task":strTask.title(), "Priority":strPrior.title()}
```

And then this row (dictionary) is added to the table (`lstTable` list). To avoid adding to the list duplicate task I applied conditional statement and if the row (dictionary) is already in the Table (list), the program informs the user and if it's not in table (list), then the program adds it to the list.

Here's the final code to add item to the list:

```
# Step 4 - Add a new item to the list/Table
elif (strChoice.strip() == '2'):
    strTask = input("Enter the task: ")
    strPrior = input("Enter Priority(Low, Medium, High): " ).title()
    tplPRIOR = ('Low', 'Medium', 'High')
    while strPrior not in tplPRIOR:
        print("Please choose Low, Medium or High")
        strPrior = input("Enter Priority(Low, Medium, High): " ).title()

    dicRow = {"Task":strTask.title(), "Priority":strPrior.title()}

    if dicRow in lstTable:
        print('Already exist.')
    else:
        lstTable.append(dicRow)
        print("\n Item added.")
    continue
```

And Figure below shows this code running in the PyCharm



```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2

Enter the task: call mom
Enter Priority(Low, Medium, High): high
Already exist.

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2

Enter the task: play games
Enter Priority(Low, Medium, High): very high
Please choose Low, Medium or High
Enter Priority(Low, Medium, High): high

 Item added.
```

```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

Call Mom, High
Cook, Medium
Walk, Low
Work Out, Medium
Play Games, High

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] -
```

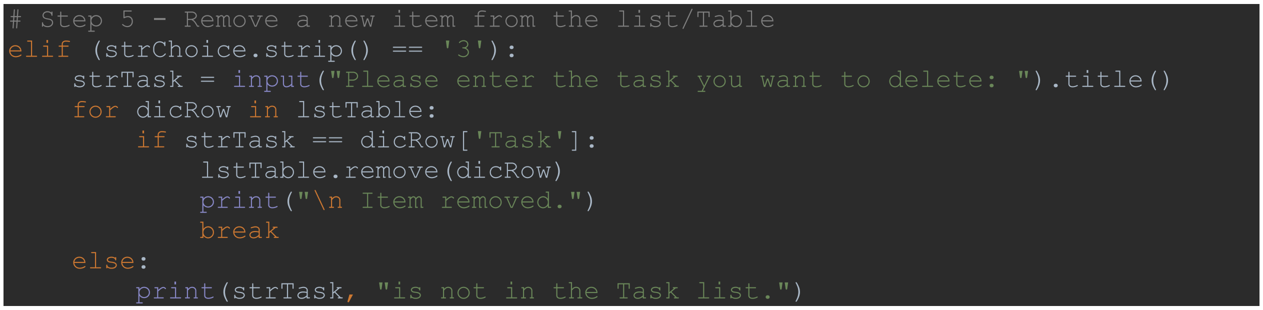
Figure 5. Screenshot of the outcome for the code to add new item

5. Remove a new item from the list/Table (Input/Output)

It’s option 3 from the menu and if the users chooses this option then the program asks to input the name of the task:

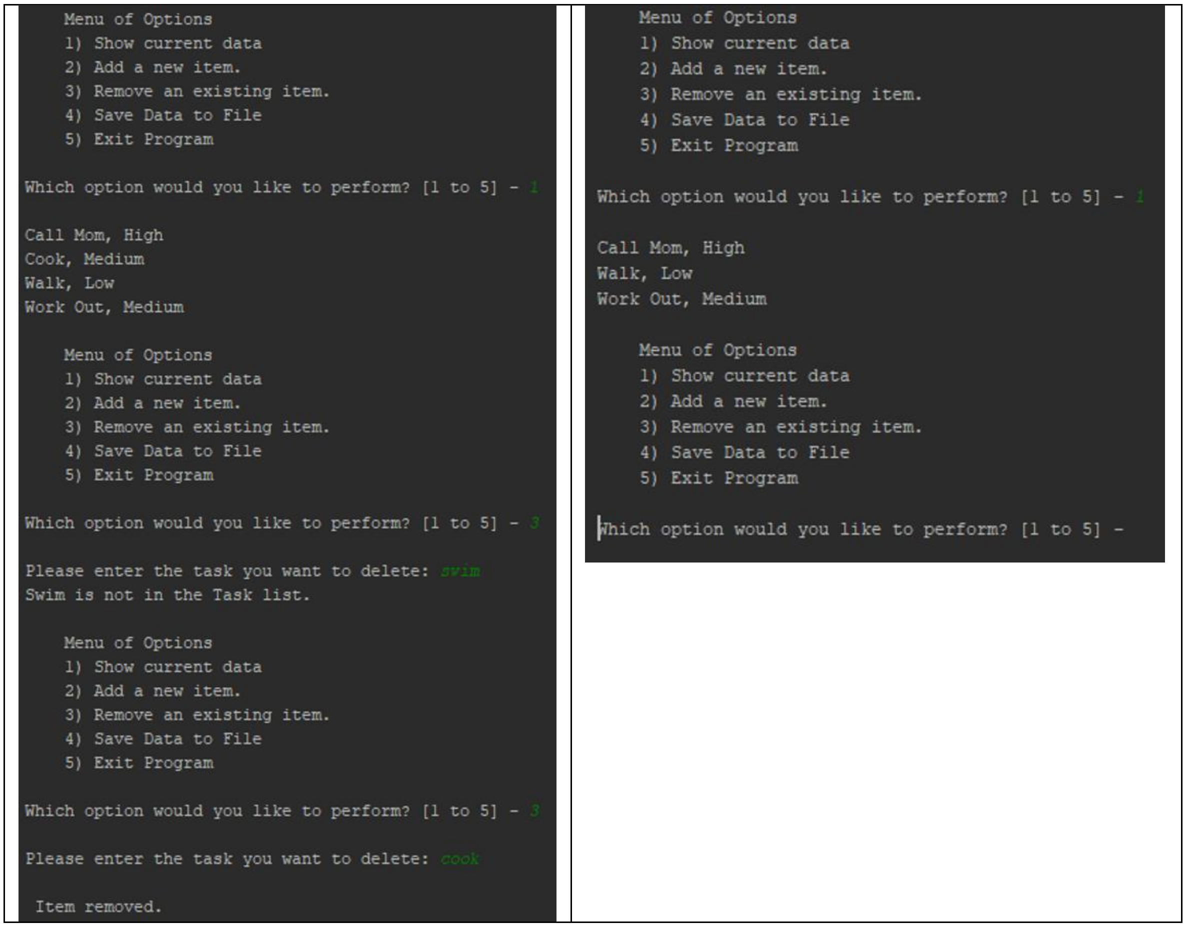
```
strTask = input("Please enter the task you want to delete: ").title()
```

Then the program goes through each row in the table and if the Task entered by the user exists in the row (which is dictionary), then `remove()` functions deletes the row, notifies the user and stops checking rows of the table. If the task is not in the table, the program notifies user. Below is the code for these steps:



```
# Step 5 - Remove a new item from the list/Table
elif (strChoice.strip() == '3'):
    strTask = input("Please enter the task you want to delete: ").title()
    for dicRow in lstTable:
        if strTask == dicRow['Task']:
            lstTable.remove(dicRow)
            print("\n Item removed.")
            break
    else:
        print(strTask, "is not in the Task list.")
```

The print screen below confirms the code works as desired.



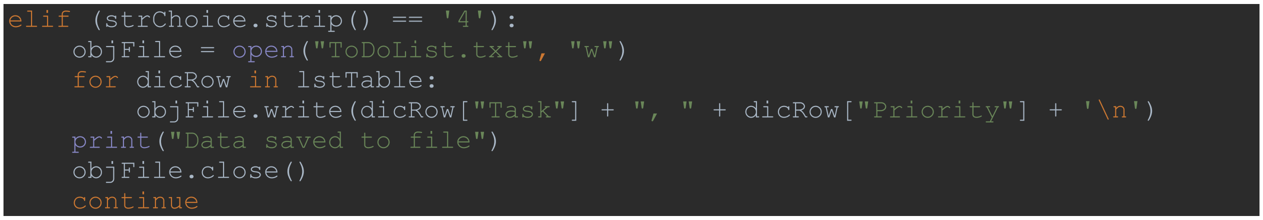
```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

Call Mom, High
Cook, Medium
Walk, Low
Work Out, Medium

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 3

Please enter the task you want to delete: swim
Swim is not in the Task list.

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 3

Please enter the task you want to delete: cook

 Item removed.
```

```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

Call Mom, High
Walk, Low
Work Out, Medium

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] -
```

Figure 6. Screenshot of the outcome for the code to remove the item from the list

6. Save tasks to the ToDolist.txt file (Input/Output section)

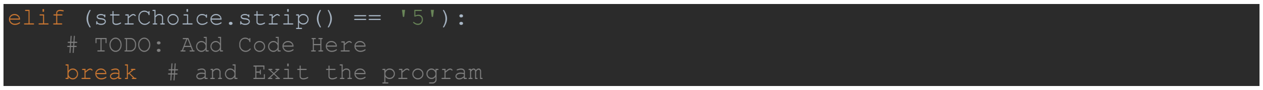
If the user chooses option 4 (Save Data to File), then the program opens the text file (function open ()), goes through each row in the table (or list) and writes each item from the row to the file (write() function). User sees notification that the data is closed and close () function closes the file.
Below is the code I wrote to save data to the file.



```
elif (strChoice.strip() == '4'):
    objFile = open("ToDoList.txt", "w")
    for dicRow in lstTable:
        objFile.write(dicRow["Task"] + ", " + dicRow["Priority"] + '\n')
    print("Data saved to file")
    objFile.close()
    continue
```

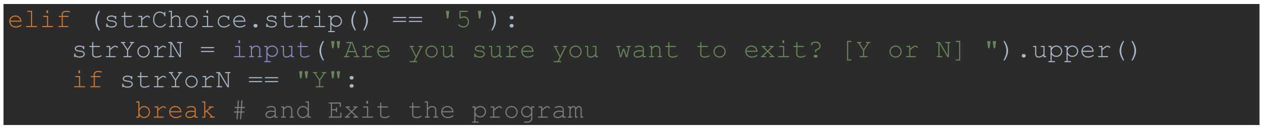
7. Exit program (Input/Output section)

It's option 5 from the Menu. And the code for this option actually works, it breaks the loop and exits if the user chooses option 5:



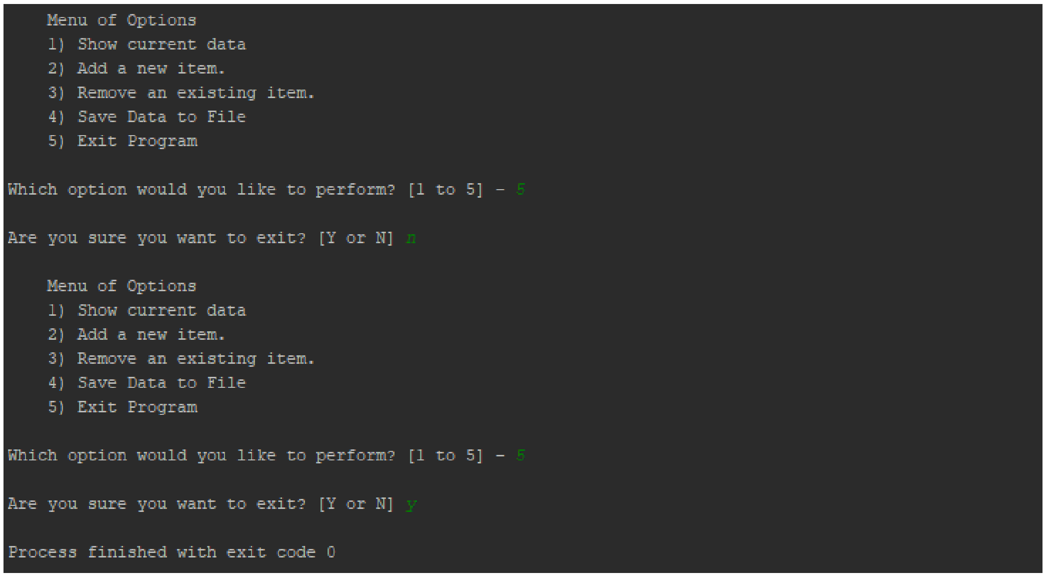
```
elif (strChoice.strip() == '5'):
    # TODO: Add Code Here
    break  # and Exit the program
```

I just added here the additional question to get confirmation from the user that he/she really wants to exit. I created the new variable `strYorN`, the value for this variable is the input from the user. If the user inputs "Y" then he exits the program. `Upper()` function allows not to worry about case sensitivity of the user's input. Any other input will tell the program to display the menu of options again.



```
elif (strChoice.strip() == '5'):
    strYorN = input("Are you sure you want to exit? [Y or N] ").upper()
    if strYorN == "Y":
        break # and Exit the program
```

And below is the print screen of the outcome if the user chooses option 5 to confirm the code is working.



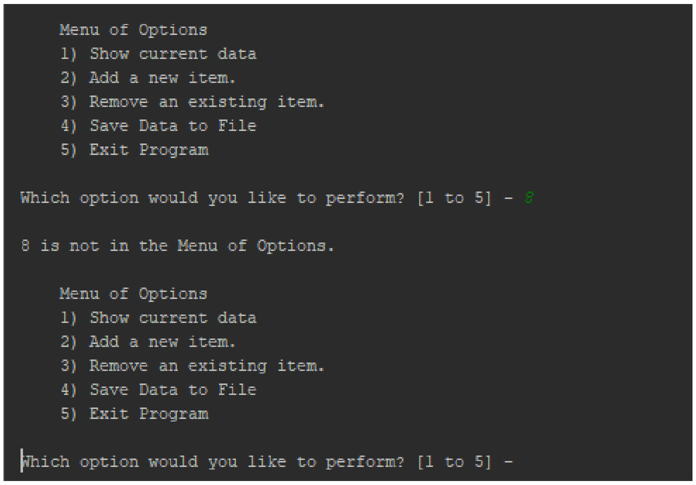
```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 5

Are you sure you want to exit? [Y or N] n

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 5

Are you sure you want to exit? [Y or N] y

Process finished with exit code 0
```

Figure 7. Screenshot of the outcome for Menu Option 5 (Exit the program)

In addition I created the final `else` condition to remind the user to choose 1, 2, 3 or 4 from the Menu of Options in case the user enters something different:

```
else:
    print(strChoice, "is not in the Menu of Options.")
```

And below is the print screen of the code outcome in case the user enters 8 which is not the Options Menu.

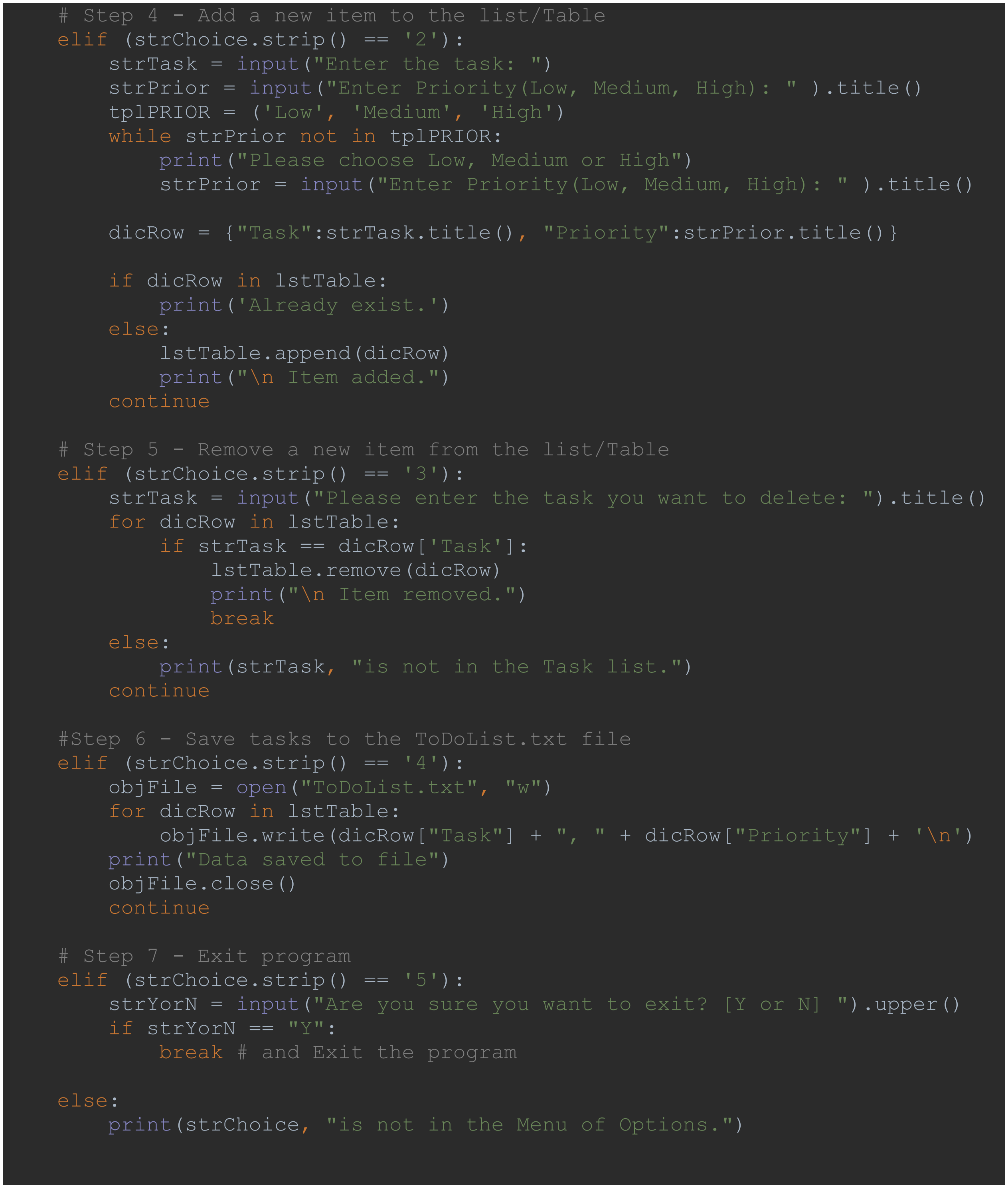


```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 8

8 is not in the Menu of Options.

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] -
```

Figure 8. Screenshot of the outcome for choosing Option that doesn't exist in the Menu.

Final Script

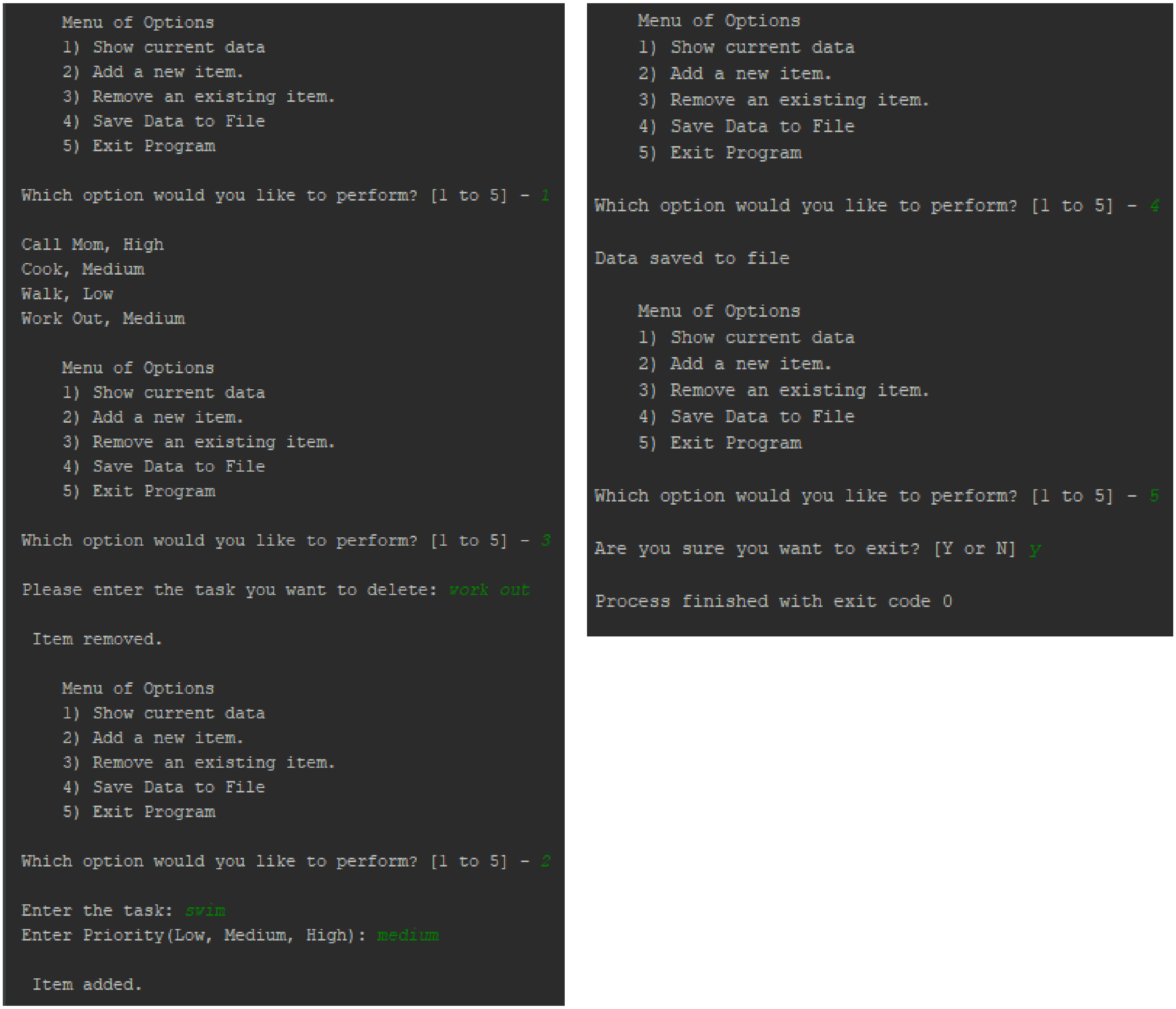
```
# ------------------------------------------------------------------------ #
# Title: Assignment 05
# Description: Working with Dictionaries and Files
#              When the program starts, load each "row" of data
#              in "ToDoToDoList.txt" into a python Dictionary.
#              Add the each dictionary "row" to a python list "table"
# ChangeLog (Who,When,What):
# RRoot,1.1.2030,Created started script
# Irina Zubova,2-14-2020,Added code to complete assignment 5
# ------------------------------------------------------------------------ #

# -- Data -- #
# declare variables and constants
objFile = "ToDoList.txt"   # An object that represents a file
strData = ""  # A row of text data from the file
dicRow = {}    # A row of data separated into elements of a dictionary
{Task,Priority}
lstTable = []  # A list that acts as a 'table' of rows
strMenu = ""   # A menu of user options
strChoice = "" # A Capture the user option selection
lstRow = []   # A list of data in the row

# -- Processing -- #
# Step 1 - When the program starts, load the any data you have
# in a text file called ToDoList.txt into a python list of dictionaries rows
(like Lab 5-2)
objFile = open("ToDoList.txt", "r")
for strData in objFile:
   lstRow = strData.split(",")
   dicRow = {"Task":lstRow[0], "Priority":lstRow[1].strip()}
   lstTable.append(dicRow)
objFile.close()


# -- Input/Output -- #
# Step 2 - Display a menu of choices to the user
while(True):
    print("""
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program
    """)
    strChoice = str(input("Which option would you like to perform? [1 to 5] -
"))
    print()  # adding a new line for looks

    # Step 3 - Show the current items in the table
    if (strChoice.strip() == '1'):
        for dicRow in lstTable:
            print(dicRow["Task"] + ", " + dicRow["Priority"].strip())
        continue
```



```
    # Step 4 - Add a new item to the list/Table
    elif (strChoice.strip() == '2'):
        strTask = input("Enter the task: ")
        strPrior = input("Enter Priority(Low, Medium, High): " ).title()
        tplPRIOR = ('Low', 'Medium', 'High')
        while strPrior not in tplPRIOR:
            print("Please choose Low, Medium or High")
            strPrior = input("Enter Priority(Low, Medium, High): " ).title()

        dicRow = {"Task":strTask.title(), "Priority":strPrior.title()}

        if dicRow in lstTable:
            print('Already exist.')
        else:
            lstTable.append(dicRow)
            print("\n Item added.")
        continue

    # Step 5 - Remove a new item from the list/Table
    elif (strChoice.strip() == '3'):
        strTask = input("Please enter the task you want to delete: ").title()
        for dicRow in lstTable:
            if strTask == dicRow['Task']:
                lstTable.remove(dicRow)
                print("\n Item removed.")
                break
        else:
            print(strTask, "is not in the Task list.")
        continue

    #Step 6 - Save tasks to the ToDoList.txt file
    elif (strChoice.strip() == '4'):
        objFile = open("ToDoList.txt", "w")
        for dicRow in lstTable:
            objFile.write(dicRow["Task"] + ", " + dicRow["Priority"] + '\n')
        print("Data saved to file")
        objFile.close()
        continue

    # Step 7 - Exit program
    elif (strChoice.strip() == '5'):
        strYorN = input("Are you sure you want to exit? [Y or N] ").upper()
        if strYorN == "Y":
            break # and Exit the program

    else:
        print(strChoice, "is not in the Menu of Options.")
```

Running the Final code

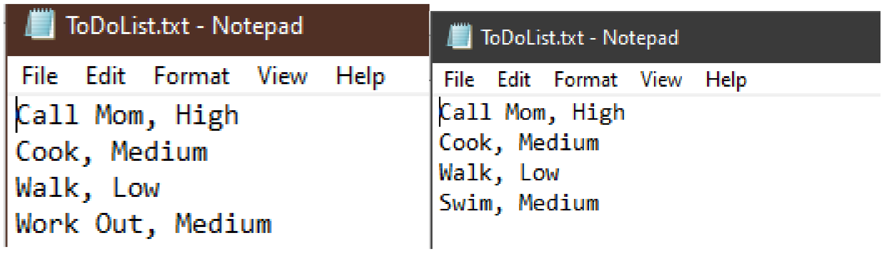
Assignment05 is asking to run the script in both OS command/shell window and PyCharm. Figure 9 captures the image of the script running in PyCharm:



```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

Call Mom, High
Cook, Medium
Walk, Low
Work Out, Medium

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 3

Please enter the task you want to delete: work out

 Item removed.

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2

Enter the task: swim
Enter Priority(Low, Medium, High): medium

 Item added.
```

```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 4

Data saved to file

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 5

Are you sure you want to exit? [Y or N] y

Process finished with exit code 0
```

Figure 9. Screenshot of the Script running in PyCharm

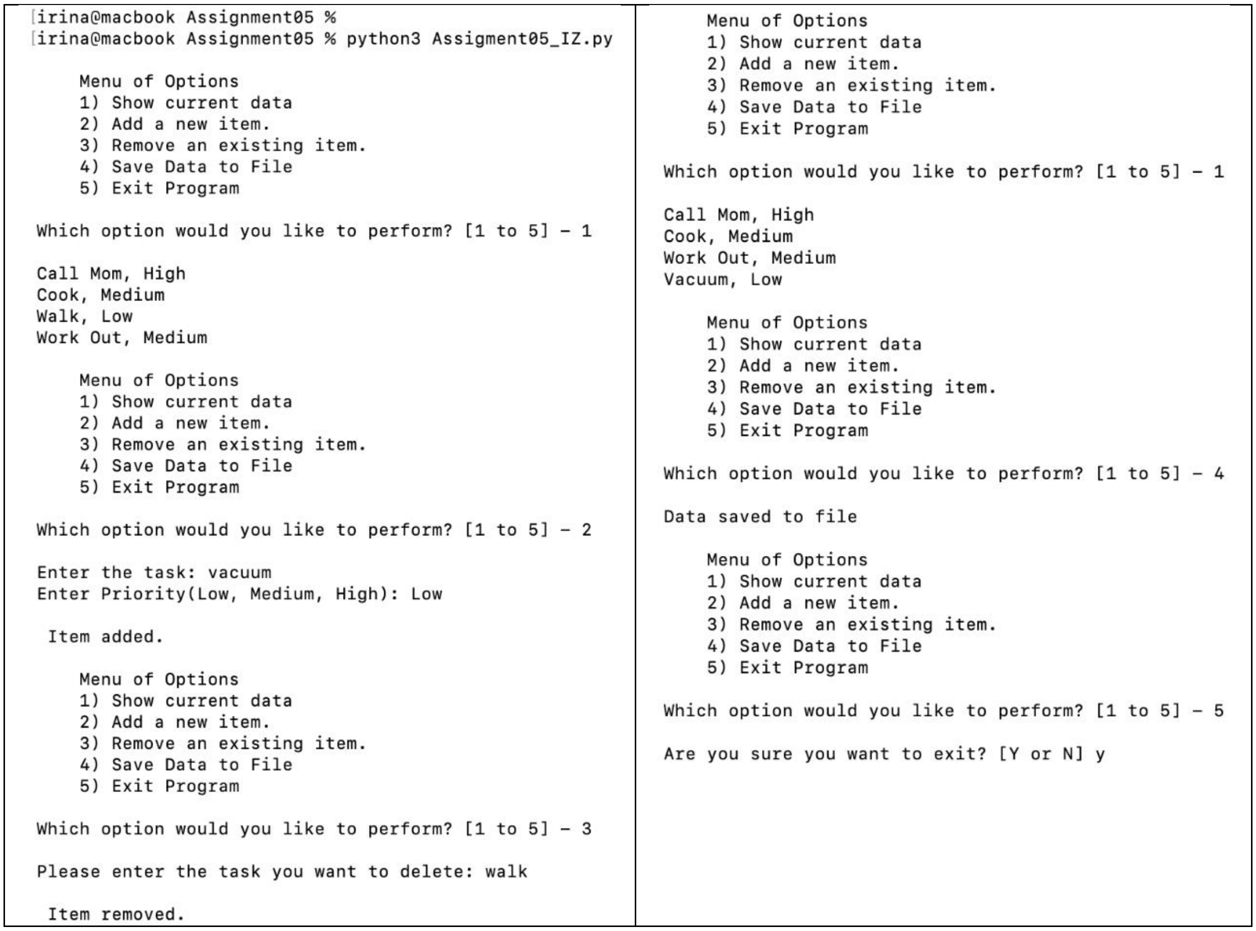


```
Call Mom, High
Cook, Medium
Walk, Low
Work Out, Medium
```

```
Call Mom, High
Cook, Medium
Walk, Low
Swim, Medium
```

Figure 10. Screenshot of the ToDo list file before (on the left) and after (on the right) the script runs

Then I ran the script in Terminal. Figure 11 shows the results:

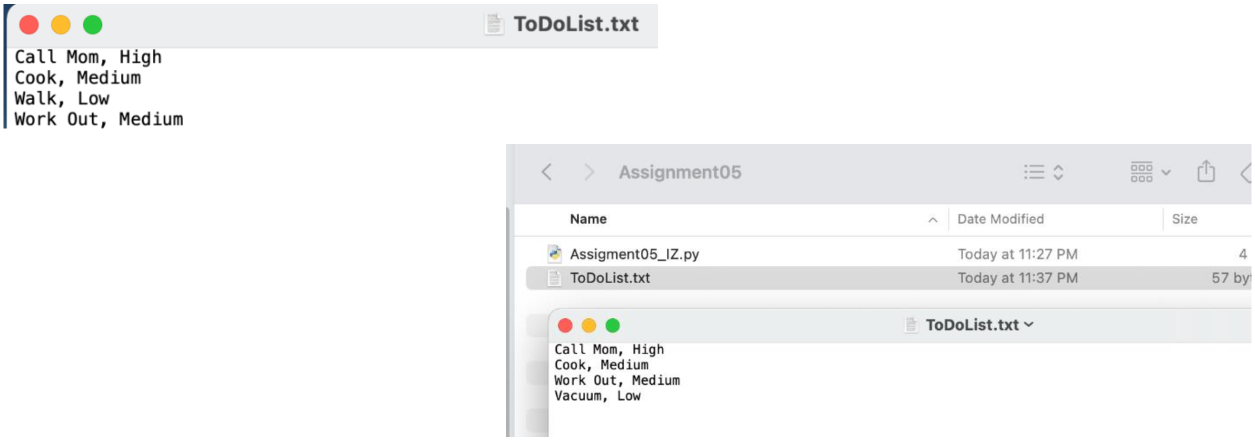


```
[irina@macbook Assignment05 %
[irina@macbook Assignment05 % python3 Assigment05_IZ.py

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

Call Mom, High
Cook, Medium
Walk, Low
Work Out, Medium

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 2

Enter the task: vacuum
Enter Priority(Low, Medium, High): Low

 Item added.

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 3

Please enter the task you want to delete: walk

 Item removed.
```

```
    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 1

Call Mom, High
Cook, Medium
Work Out, Medium
Vacuum, Low

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 4

Data saved to file

    Menu of Options
    1) Show current data
    2) Add a new item.
    3) Remove an existing item.
    4) Save Data to File
    5) Exit Program

Which option would you like to perform? [1 to 5] - 5

Are you sure you want to exit? [Y or N] y
```

Figure 11. Screenshot of the Script running in Command Window

And the last picture (Figure 12) verifies the data in the file was changed according to the user inputs. The original text file is in the upper left corner, the final version after user's changes in in the lower right corner.



ToDoList.txt

```
Call Mom, High
Cook, Medium
Walk, Low
Work Out, Medium
```

Name	Date Modified	Size
Assigment05_IZ.py	Today at 11:27 PM	4
ToDoList.txt	Today at 11:37 PM	57 by

ToDoList.txt

```
Call Mom, High
Cook, Medium
Work Out, Medium
Vacuum, Low
```

Figure 12. Print screens of the text file before and after running the script in Command Window

Summary

In this assignment practiced more in working with the list sequences in Python, using `while` and `for` loops, creating multiple conditions. I learned how to create and modify dictionaries in Python, how to access specific items in the dictionaries. I was able to open the text file in Python, process the data from this file to represent it in a virtual table where the table is a list and rows of the table are dictionaries. I was able to make changes in this table like adding and deleting rows and then I finally saved the updated data back to the text file.