Carol Hubbard
November 13, 2022
Foundations Of Programming: Python
Assignment 05
Github URL: https://github.com/CarolHubbard/IntroToProg-Python

To-Do List Script

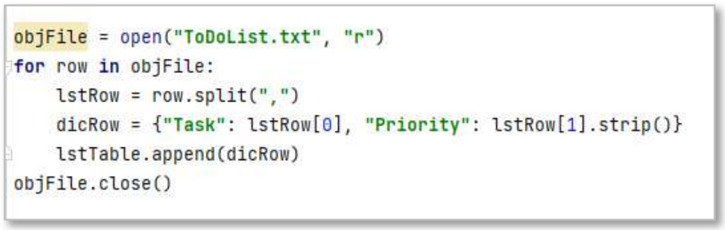
Introduction

The "`To-Do List`" script looks at the operations of a simple To-Do List. This script allows you to add or remove items from the list, view existing items on the list or save the items in the list to a text file.

It works by prompting the user for two pieces of information: a task and the priority assigned to it. This data is then placed into a list in the form of dictionary rows. The user is then presented with a menu of options that will allow them to manipulate the data in the list.

The List

The script begins by loading the contents of a text file ("`ToDoList.txt`") into a list of dictionary rows as seen in **Figure 1.** This is where the items of the To-Do List will be stored.

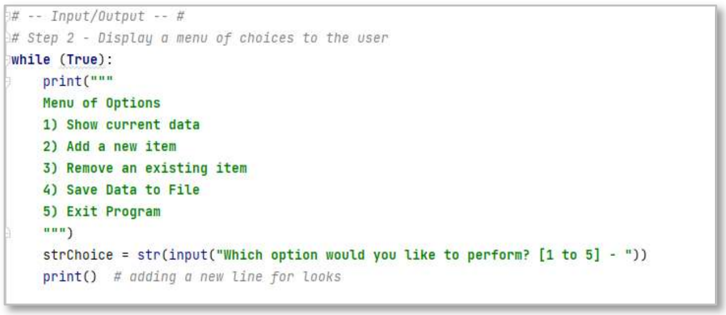


```
objFile = open("ToDoList.txt", "r")
for row in objFile:
    lstRow = row.split(",")
    dicRow = {"Task": lstRow[0], "Priority": lstRow[1].strip()}
    lstTable.append(dicRow)
objFile.close()
```

Figure 1 Loads data from .txt file into list

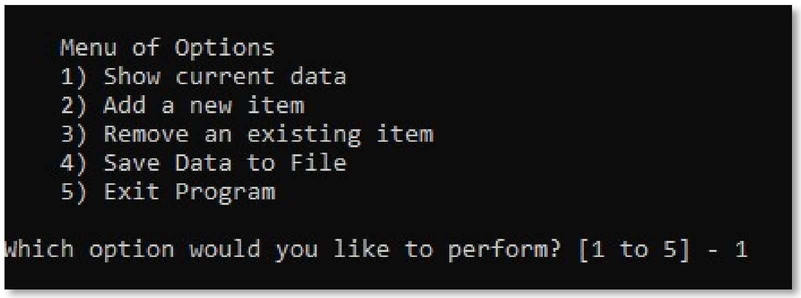
The Menu

As noted previously, the user is presented with a menu of options to choose from. This menu is running inside a `While` loop and will continue to display to the user until they press `5` to exit the program.



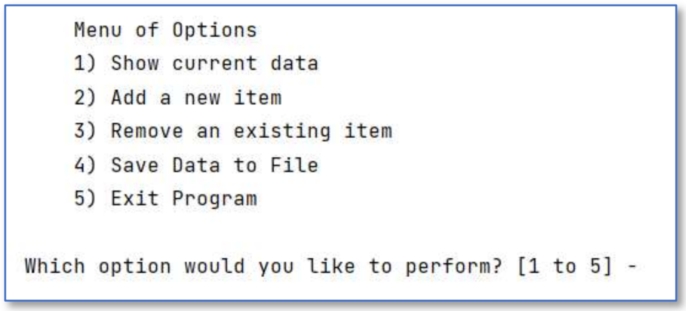
```
# -- Input/Output -- #
# Step 2 - Display a menu of choices to the user
while (True):
    print("""
    Menu of Options
    1) Show current data
    2) Add a new item
    3) Remove an existing item
    4) Save Data to File
    5) Exit Program
    """)
    strChoice = str(input("Which option would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

Figure 2 While loop of 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] - 1
```

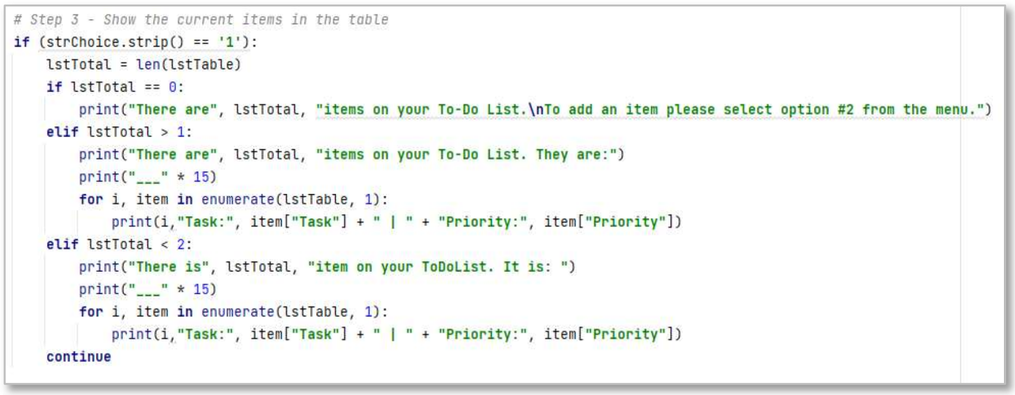
Figure 3 Menu options presented in command window



```
    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 Menu options as seen in PyCharm

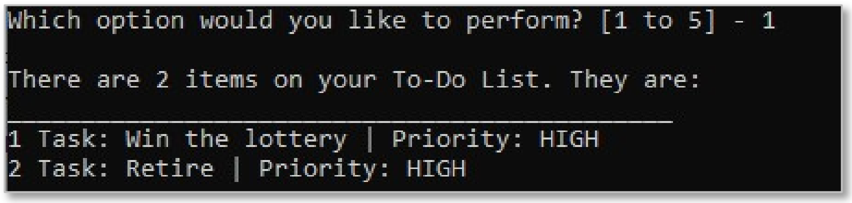
If the user chooses **Option 1**, the current contents of the text file will be displayed to the user as a numbered list for their review.



```python
# Step 3 - Show the current items in the table
if (strChoice.strip() == '1'):
    lstTotal = len(lstTable)
    if lstTotal == 0:
        print("There are", lstTotal, "items on your To-Do List.\nTo add an item please select option #2 from the menu.")
    elif lstTotal > 1:
        print("There are", lstTotal, "items on your To-Do List. They are:")
        print("___" * 15)
        for i, item in enumerate(lstTable, 1):
            print(i,"Task:", item["Task"] + " | " + "Priority:", item["Priority"])
    elif lstTotal < 2:
        print("There is", lstTotal, "item on your ToDoList. It is: ")
        print("___" * 15)
        for i, item in enumerate(lstTable, 1):
            print(i,"Task:", item["Task"] + " | " + "Priority:", item["Priority"])
    continue
```

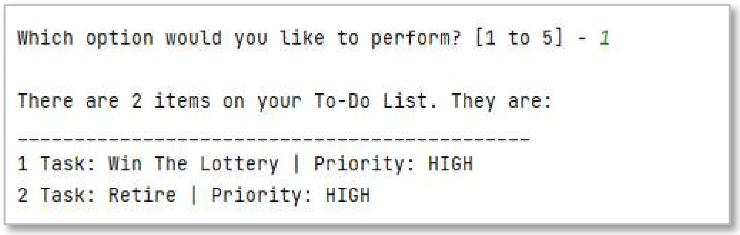
Figure 5 Code to display items currently in To-Do List

The numbering was achieved through the use of the built-in `enumerate` function which was applied to each row in `lstTable`.



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

There are 2 items on your To-Do List. They are:
_____________________________________________
1 Task: Win the lottery | Priority: HIGH
2 Task: Retire | Priority: HIGH
```

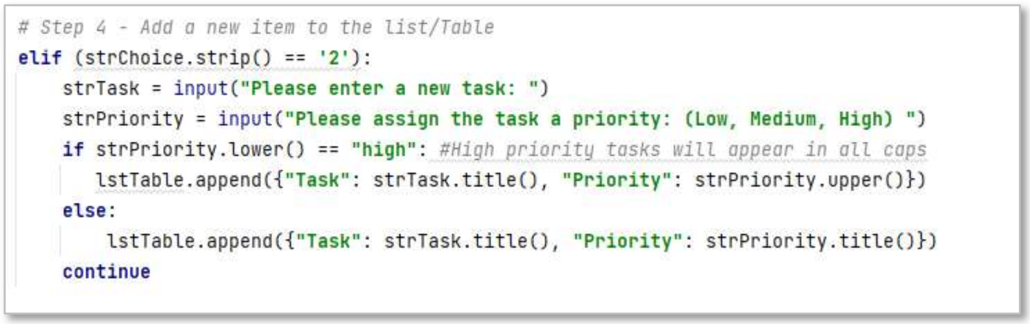
Figure 6 List of items in To-Do list



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

There are 2 items on your To-Do List. They are:
----------------------------------------------
1 Task: Win The Lottery | Priority: HIGH
2 Task: Retire | Priority: HIGH
```

Figure 7 List of items using PyCharm

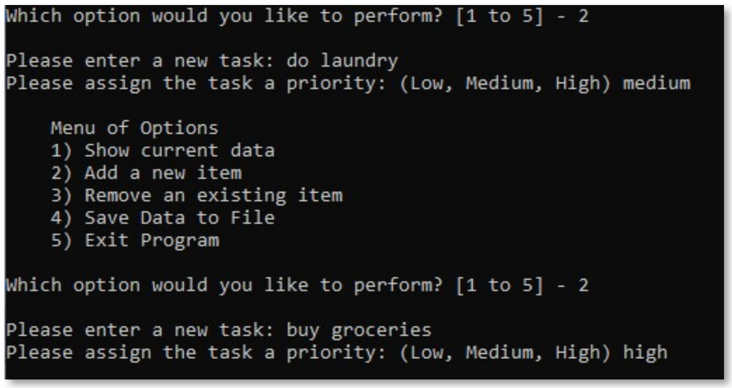
Choosing **Option 2** allows the user to add an item to their To-Do List and assign a priority of either low, medium, or high.



```
# Step 4 - Add a new item to the list/Table
elif (strChoice.strip() == '2'):
    strTask = input("Please enter a new task: ")
    strPriority = input("Please assign the task a priority: (Low, Medium, High) ")
    if strPriority.lower() == "high": #High priority tasks will appear in all caps
       lstTable.append({"Task": strTask.title(), "Priority": strPriority.upper()})
    else:
        lstTable.append({"Task": strTask.title(), "Priority": strPriority.title()})
    continue
```

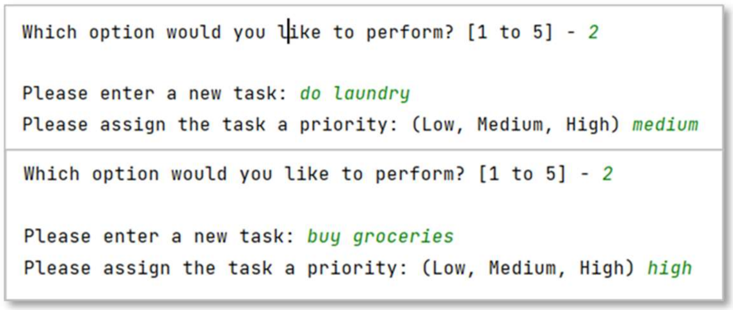
Figure 8 Option 2 code to add task to list

I entered two new items: do laundry and buy groceries as seen in **Figures 9** and **10**.



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

Please enter a new task: do laundry
Please assign the task a priority: (Low, Medium, High) 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

Please enter a new task: buy groceries
Please assign the task a priority: (Low, Medium, High) high
```

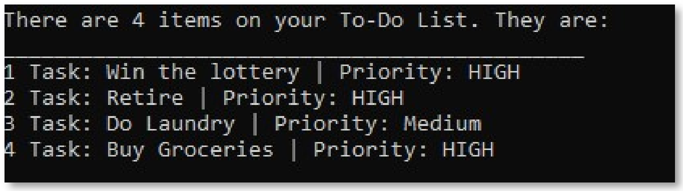
Figure 9 Adding item to To-Do List



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

Please enter a new task: do laundry
Please assign the task a priority: (Low, Medium, High) medium
Which option would you like to perform? [1 to 5] - 2

Please enter a new task: buy groceries
Please assign the task a priority: (Low, Medium, High) high
```

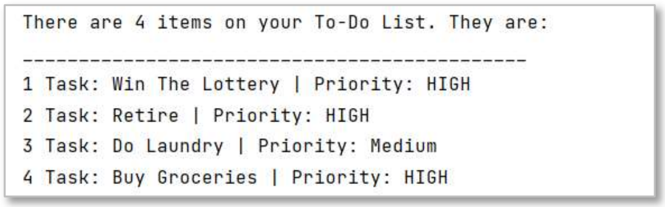
Figure 10 Output in PyCharm

Notice how the output is formatted. The tasks and priorities were both entered in all lowercase; however, the output has been changed so that priority for high priority tasks is displayed in all caps and all other data has been changed to title case. This is done using the `.upper` and `.title` methods respectively.



```
There are 4 items on your To-Do List. They are:
_____________________________________________
1 Task: Win the lottery | Priority: HIGH
2 Task: Retire | Priority: HIGH
3 Task: Do Laundry | Priority: Medium
4 Task: Buy Groceries | Priority: HIGH
```

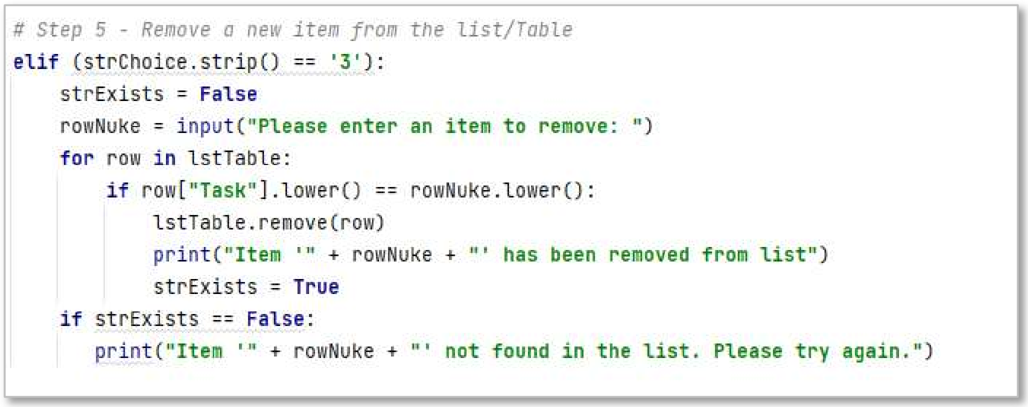
Figure 11 Numbered To-Do List



```
There are 4 items on your To-Do List. They are:
---------------------------------------------
1 Task: Win The Lottery | Priority: HIGH
2 Task: Retire | Priority: HIGH
3 Task: Do Laundry | Priority: Medium
4 Task: Buy Groceries | Priority: HIGH
```

Figure 12 Numbered list in PyCharm

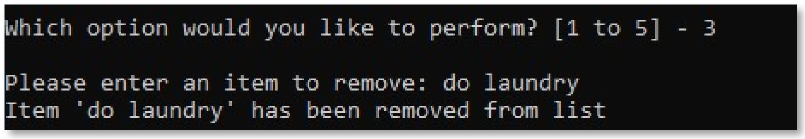
To remove items from the list, the user selects **Option 3**. The user is prompted to enter the item they want to remove and then the script will loop through each item in the list to see if it exists. If it does, the item will be removed and the user will be notified. Otherwise an error will be generated and the user will be prompted to try again.



```python
# Step 5 - Remove a new item from the list/Table
elif (strChoice.strip() == '3'):
    strExists = False
    rowNuke = input("Please enter an item to remove: ")
    for row in lstTable:
        if row["Task"].lower() == rowNuke.lower():
            lstTable.remove(row)
            print("Item '" + rowNuke + "' has been removed from list")
            strExists = True
    if strExists == False:
        print("Item '" + rowNuke + "' not found in the list. Please try again.")
```

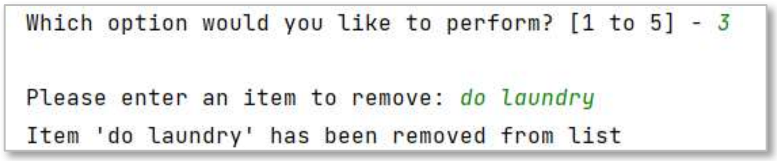
Figure 13 Code to delete an item from the list

I ***really*** don't like doing laundry, so I think I'll remove that from the list.



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

Please enter an item to remove: do laundry
Item 'do laundry' has been removed from list
```

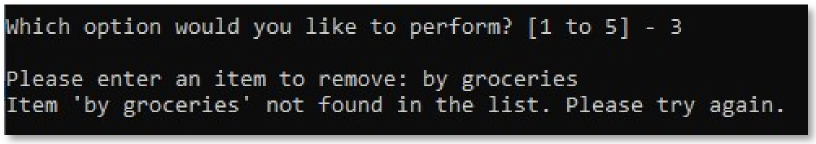
Figure 14 Deletes an item from the list



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

Please enter an item to remove: do laundry
Item 'do laundry' has been removed from list
```

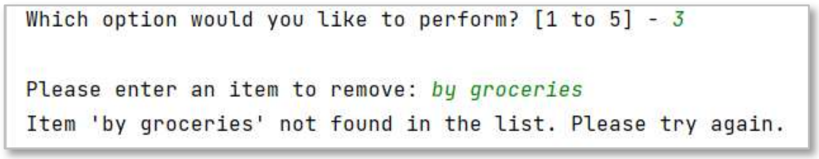
Figure 15 Deleting an item from the list - PyCharm

Grocery shopping can probably wait too. I'll remove it as well.



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

Please enter an item to remove: by groceries
Item 'by groceries' not found in the list. Please try again.
```

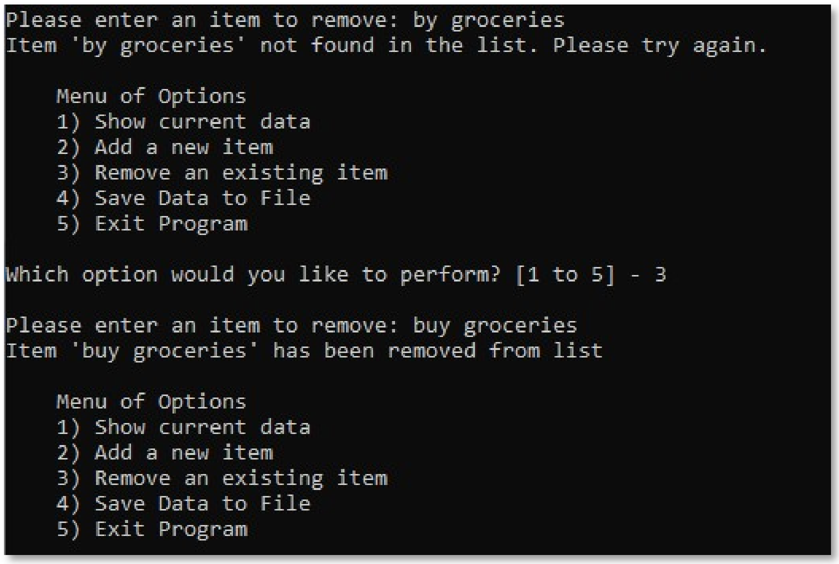
Figure 16 Deleting an item that doesn't exist in the list



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

Please enter an item to remove: by groceries
Item 'by groceries' not found in the list. Please try again.
```

Figure 17 Deleting an item that doesn't exist in the list

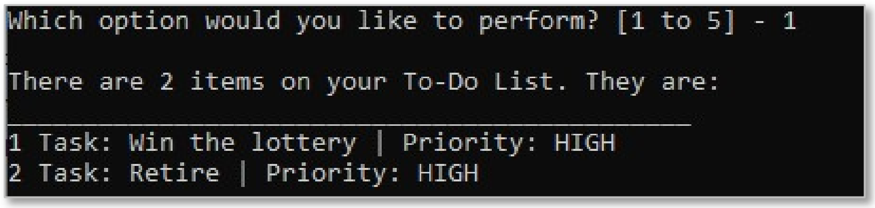
This generated an error because "`buy`" was misspelled "`by`" and there is no item on my To-Do List that matches that. Trying again spelling everything correctly works as expected:



```
Please enter an item to remove: by groceries
Item 'by groceries' not found in the list. Please try again.

    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 an item to remove: buy groceries
Item 'buy groceries' has been removed from 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
```

Figure 18 Correcting typo and deleting item

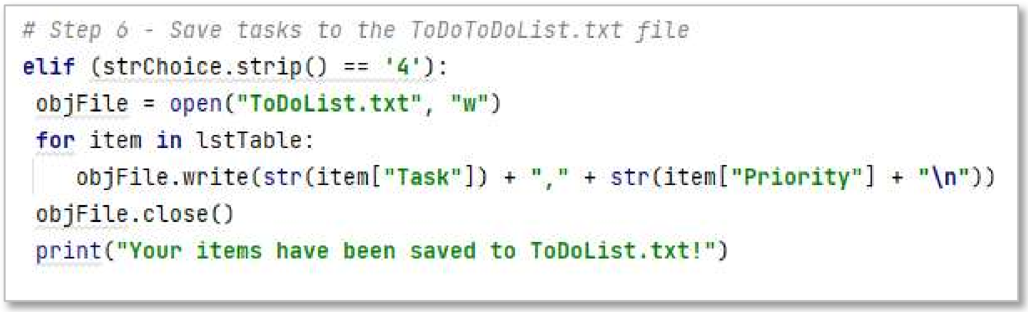
There, that's better. With those two items removed, my To-Do List is much shorter.



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

There are 2 items on your To-Do List. They are:
______________________________________________
1 Task: Win the lottery | Priority: HIGH
2 Task: Retire | Priority: HIGH
```

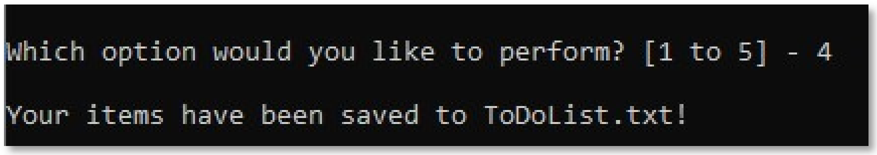
Figure 19 Updated list after items removed

Finally, when the user is satisfied with their updated To-Do List, they can save the list to a text file by choosing **Option 4**:



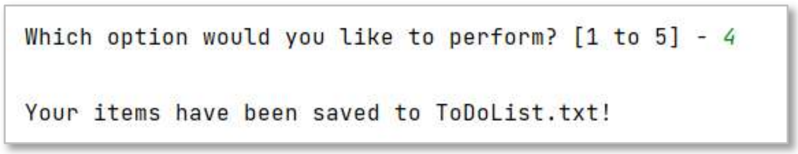
```python
# Step 6 - Save tasks to the ToDoToDoList.txt file
elif (strChoice.strip() == '4'):
    objFile = open("ToDoList.txt", "w")
    for item in lstTable:
        objFile.write(str(item["Task"]) + "," + str(item["Priority"] + "\n"))
    objFile.close()
    print("Your items have been saved to ToDoList.txt!")
```

Figure 20 Code that saves list items to text file



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

Your items have been saved to ToDoList.txt!
```

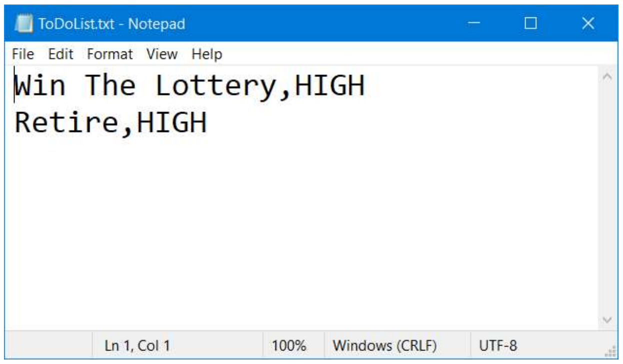
Figure 21 Selecting option 4



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

Your items have been saved to ToDoList.txt!
```

Figure 22 Saving list items to text file in PyCharm

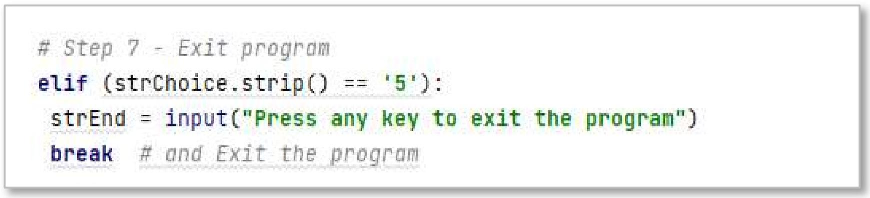
Figure 23 shows the items from the To-Do List were successfully written to `ToDoList.txt`.



```
Win The Lottery,HIGH
Retire,HIGH
```

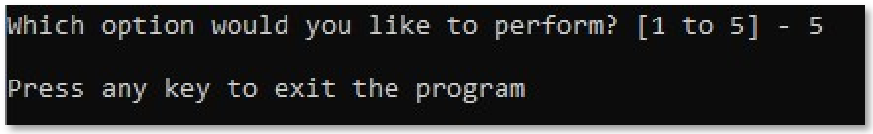
Figure 23 ToDoList.txt with tasks and priorities

To exit out of the script, simply press **Option 5**, then press any key.



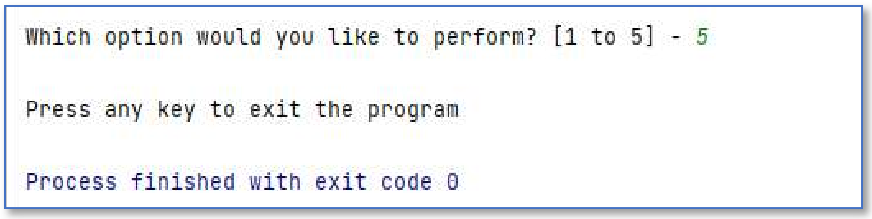
```
# Step 7 - Exit program
elif (strChoice.strip() == '5'):
 strEnd = input("Press any key to exit the program")
 break  # and Exit the program
```

Figure 24 Exits the script and closes



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

Press any key to exit the program
```

Figure 25 User presses 5 to exit the script



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

Press any key to exit the program

Process finished with exit code 0
```

Figure 26 PyCharm user exiting script

Summary

This script demonstrates the use of dictionaries and lists to create a standard To-Do List. It shows how, through the use of `Elif` statements, you can manipulate the data in the list to add, remove, view, or save the data to a text file.