Paula Shoup

November 13, 2019

Foundations of Programming: Python

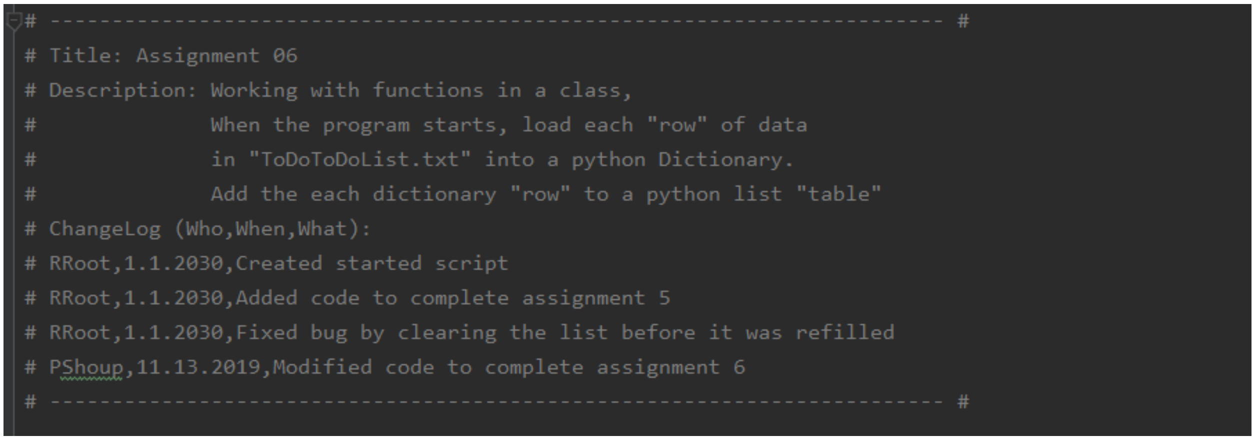
Assignment 06

Converting a Task List Menu using Functions

Converting a Task List using Functions

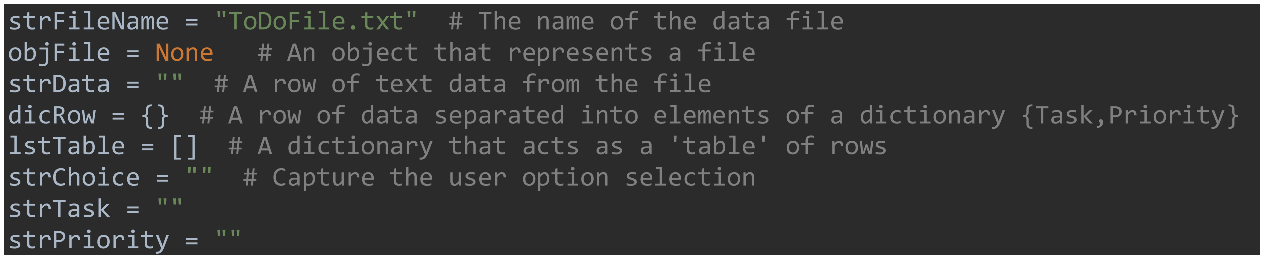
The below script contains instructions converted from its original code into instructions. The script will display a menu of options that allows the user to enter a task and its priority level, display the tasks the user entered, allow the user to remove an item, save items, and reload items before exiting the program. If the user opts to save their data the program will save the information to a text file called ToDoList.txt.

To create the script, start by opening the starter script file and updating the ChangeLog.



```
# ---------------------------------------------------------------------- #
# Title: Assignment 06
# Description: Working with functions in a class,
#              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
# RRoot,1.1.2030,Added code to complete assignment 5
# RRoot,1.1.2030,Fixed bug by clearing the list before it was refilled
# PShoup,11.13.2019,Modified code to complete assignment 6
# ---------------------------------------------------------------------- #
```

This script will create eight variables:

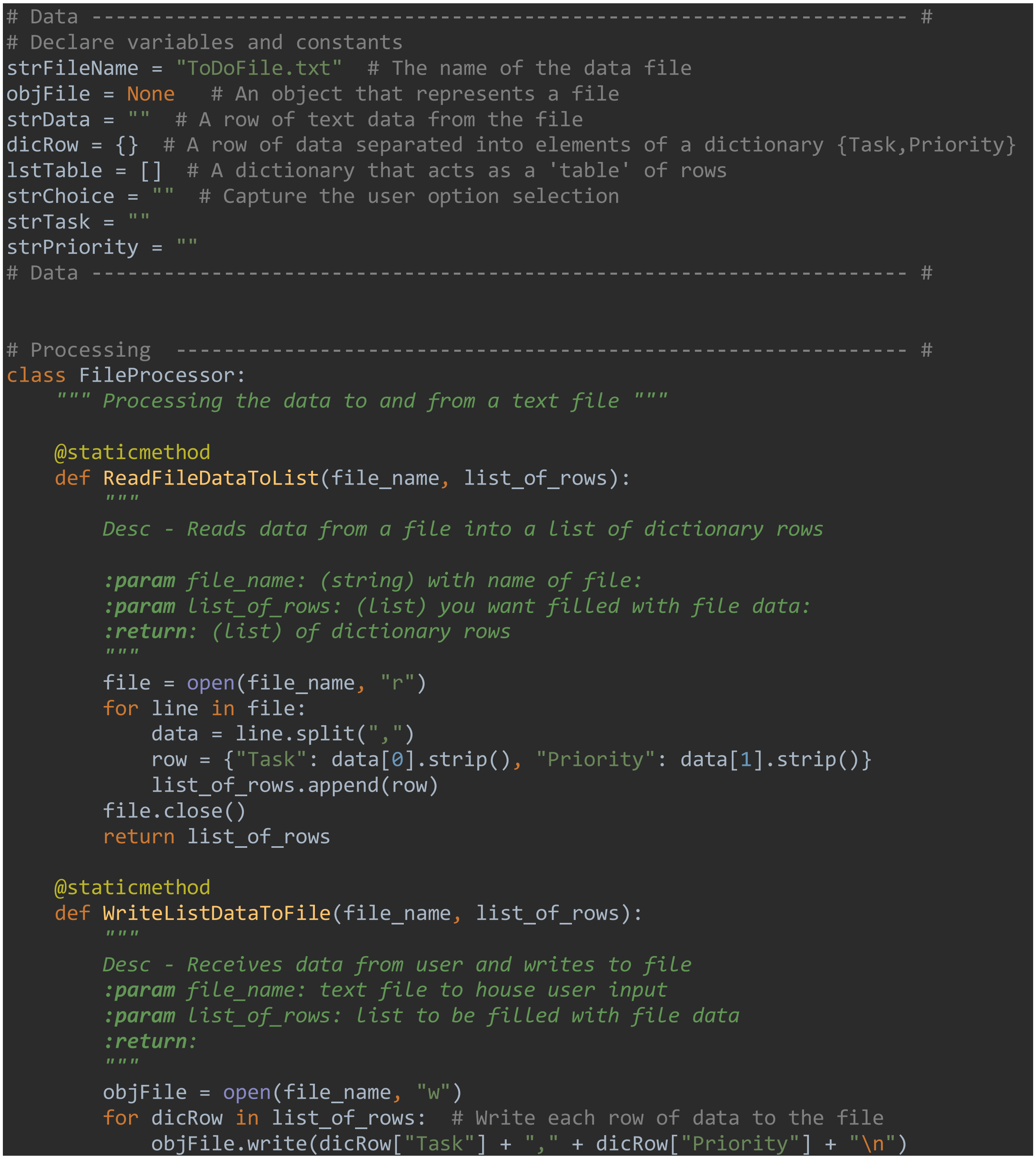


```
strFileName = "ToDoFile.txt"  # The name of the data file
objFile = None   # An object that represents a file
strData = ""  # A row of text data from the file
dicRow = {}  # A row of data separated into elements of a dictionary {Task,Priority}
lstTable = []  # A dictionary that acts as a 'table' of rows
strChoice = ""  # Capture the user option selection
strTask = ""
strPriority = ""
```

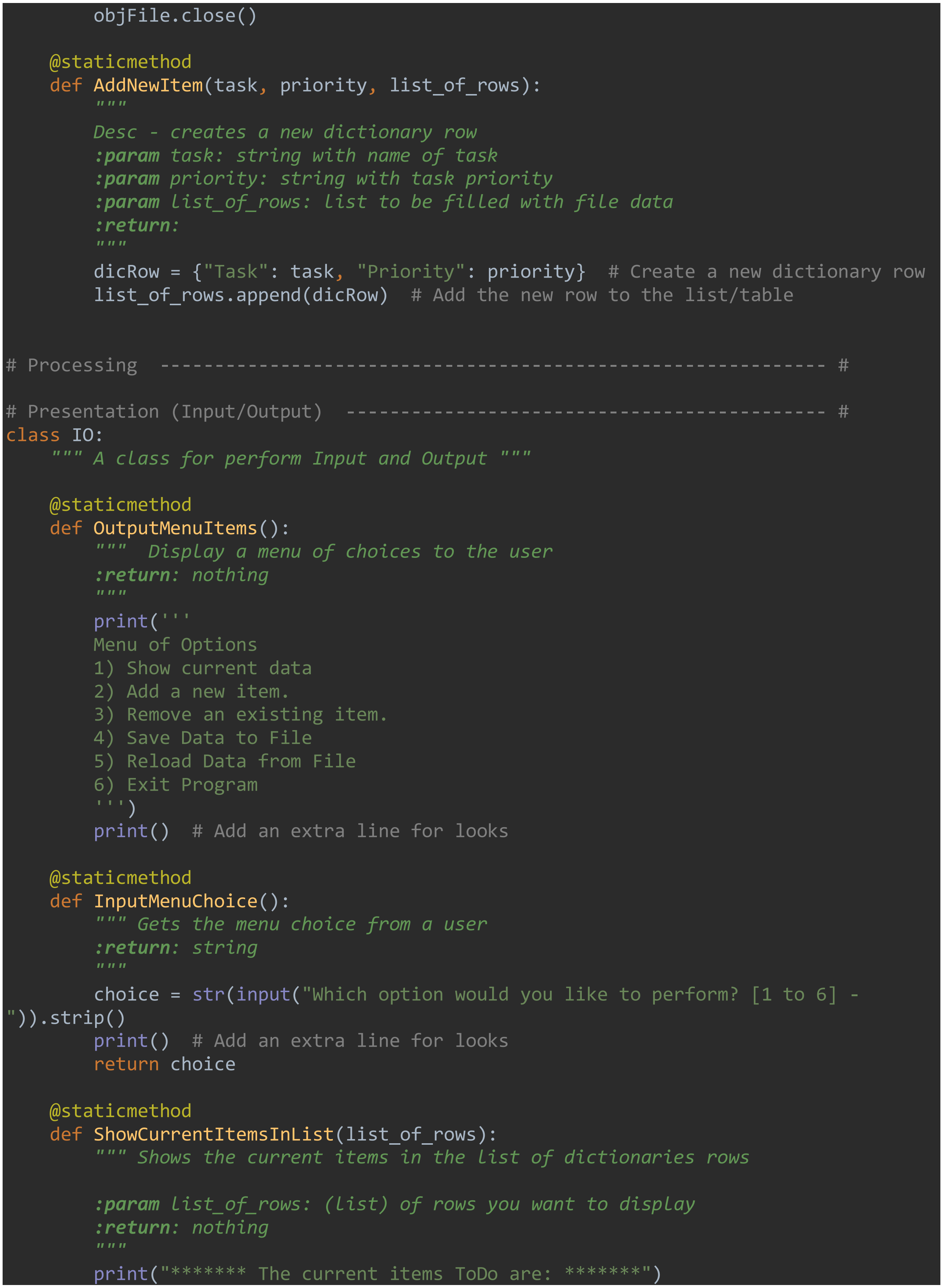
The script uses six functions; print(), input(), append(), open(), write(), close(). The script evaluates the user’s selection by running through a series of while, elif statements that will either receive user input, display data, delete data, reload data, or save data to file and exit.

Below is the script that will provide the user with a set of options to enter, display, delete, reload, or save inventory data. At the direction of the user, the program will save the data to a newly created text file, print a message to confirm the data has been saved. The user can also opt to exit the program at any point from the main menu.

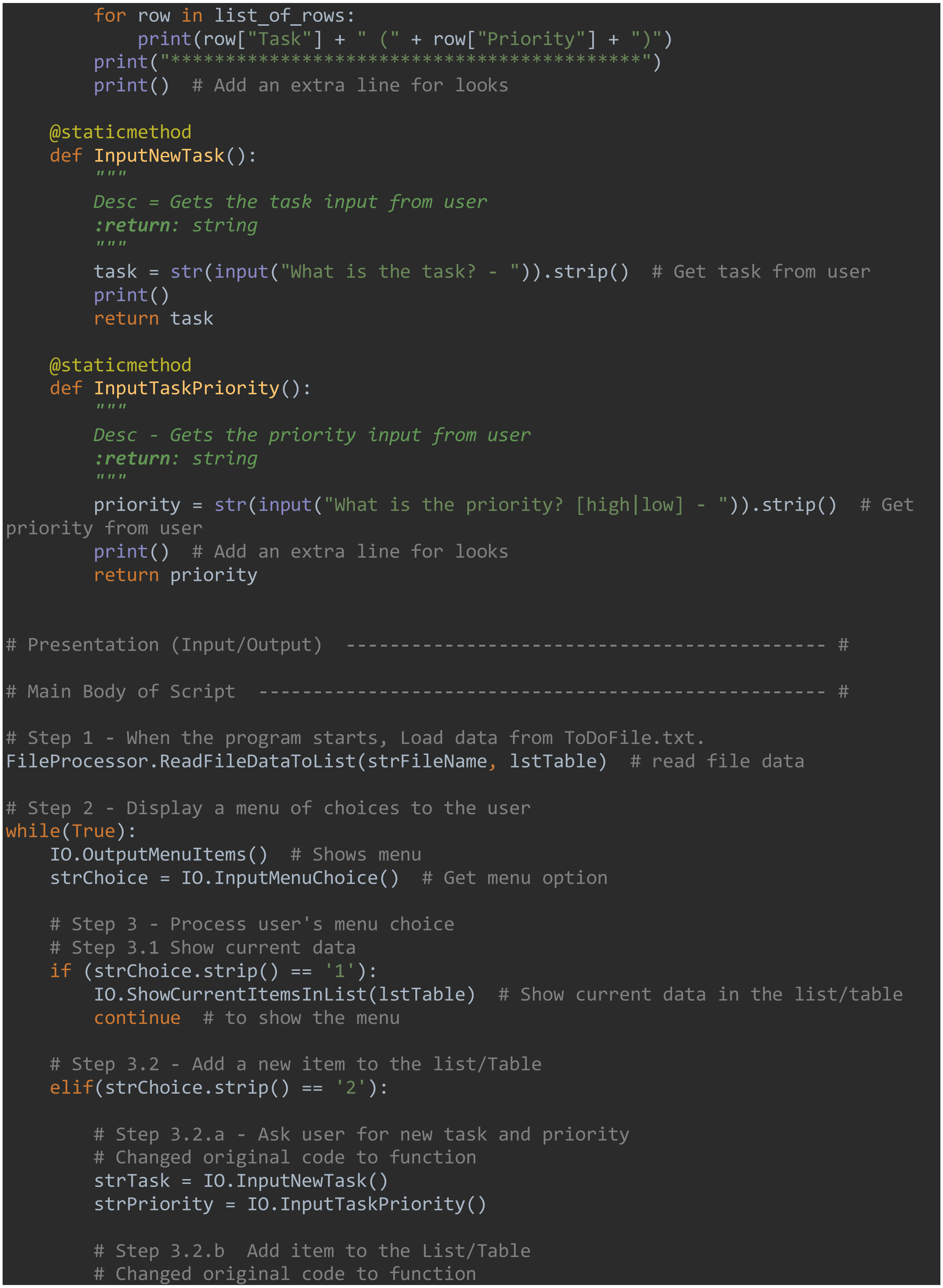
(The script should be saved as “Assignment06.py” in the Assignment06 folder within _PythonClass in the C: drive.)



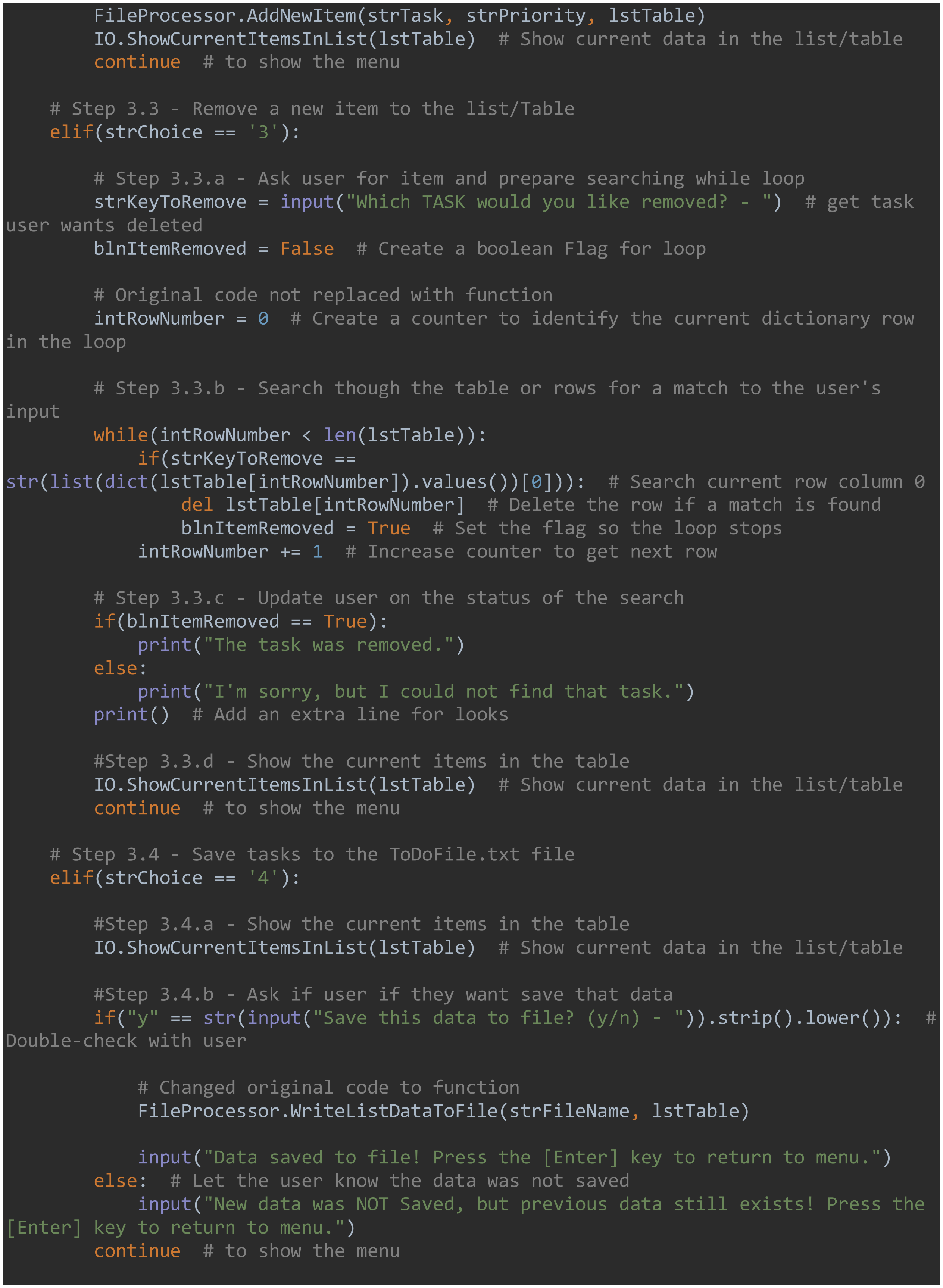
```python
# Data -------------------------------------------------------------------- #
# Declare variables and constants
strFileName = "ToDoFile.txt"  # The name of the data file
objFile = None   # An object that represents a file
strData = ""  # A row of text data from the file
dicRow = {}  # A row of data separated into elements of a dictionary {Task,Priority}
lstTable = []  # A dictionary that acts as a 'table' of rows
strChoice = ""  # Capture the user option selection
strTask = ""
strPriority = ""
# Data -------------------------------------------------------------------- #


# Processing  ------------------------------------------------------------- #
class FileProcessor:
    """ Processing the data to and from a text file """

    @staticmethod
    def ReadFileDataToList(file_name, list_of_rows):
        """
        Desc - Reads data from a file into a list of dictionary rows

        :param file_name: (string) with name of file:
        :param list_of_rows: (list) you want filled with file data:
        :return: (list) of dictionary rows
        """
        file = open(file_name, "r")
        for line in file:
            data = line.split(",")
            row = {"Task": data[0].strip(), "Priority": data[1].strip()}
            list_of_rows.append(row)
        file.close()
        return list_of_rows

    @staticmethod
    def WriteListDataToFile(file_name, list_of_rows):
        """
        Desc - Receives data from user and writes to file
        :param file_name: text file to house user input
        :param list_of_rows: list to be filled with file data
        :return:
        """
        objFile = open(file_name, "w")
        for dicRow in list_of_rows:  # Write each row of data to the file
            objFile.write(dicRow["Task"] + "," + dicRow["Priority"] + "\n")
```



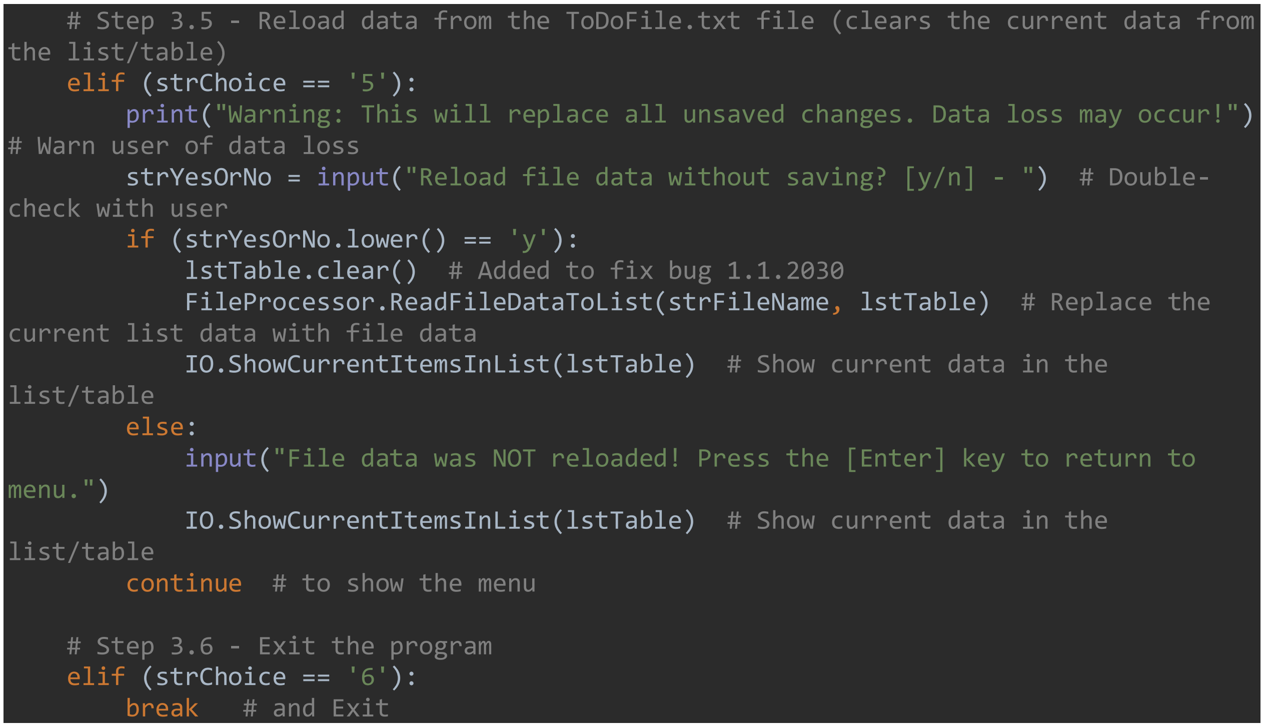
```
        objFile.close()

    @staticmethod
    def AddNewItem(task, priority, list_of_rows):
        """
        Desc - creates a new dictionary row
        :param task: string with name of task
        :param priority: string with task priority
        :param list_of_rows: list to be filled with file data
        :return:
        """
        dicRow = {"Task": task, "Priority": priority}  # Create a new dictionary row
        list_of_rows.append(dicRow)  # Add the new row to the list/table


# Processing  ------------------------------------------------------------- #

# Presentation (Input/Output)  -------------------------------------------- #
class IO:
    """ A class for perform Input and Output """

    @staticmethod
    def OutputMenuItems():
        """  Display a menu of choices to the user
        :return: nothing
        """
        print('''
        Menu of Options
        1) Show current data
        2) Add a new item.
        3) Remove an existing item.
        4) Save Data to File
        5) Reload Data from File
        6) Exit Program
        ''')
        print()  # Add an extra line for looks

    @staticmethod
    def InputMenuChoice():
        """ Gets the menu choice from a user
        :return: string
        """
        choice = str(input("Which option would you like to perform? [1 to 6] -
")).strip()
        print()  # Add an extra line for looks
        return choice

    @staticmethod
    def ShowCurrentItemsInList(list_of_rows):
        """ Shows the current items in the list of dictionaries rows

        :param list_of_rows: (list) of rows you want to display
        :return: nothing
        """
        print("******* The current items ToDo are: *******")
```



```
        for row in list_of_rows:
            print(row["Task"] + " (" + row["Priority"] + ")")
        print("*******************************************")
        print()  # Add an extra line for looks

    @staticmethod
    def InputNewTask():
        """
        Desc = Gets the task input from user
        :return: string
        """
        task = str(input("What is the task? - ")).strip()  # Get task from user
        print()
        return task

    @staticmethod
    def InputTaskPriority():
        """
        Desc - Gets the priority input from user
        :return: string
        """
        priority = str(input("What is the priority? [high|low] - ")).strip()  # Get
priority from user
        print()  # Add an extra line for looks
        return priority


# Presentation (Input/Output)  -------------------------------------------- #

# Main Body of Script  ---------------------------------------------------- #

# Step 1 - When the program starts, Load data from ToDoFile.txt.
FileProcessor.ReadFileDataToList(strFileName, lstTable)  # read file data

# Step 2 - Display a menu of choices to the user
while(True):
    IO.OutputMenuItems()  # Shows menu
    strChoice = IO.InputMenuChoice()  # Get menu option

    # Step 3 - Process user's menu choice
    # Step 3.1 Show current data
    if (strChoice.strip() == '1'):
        IO.ShowCurrentItemsInList(lstTable)  # Show current data in the list/table
        continue  # to show the menu

    # Step 3.2 - Add a new item to the list/Table
    elif(strChoice.strip() == '2'):

        # Step 3.2.a - Ask user for new task and priority
        # Changed original code to function
        strTask = IO.InputNewTask()
        strPriority = IO.InputTaskPriority()

        # Step 3.2.b  Add item to the List/Table
        # Changed original code to function
```

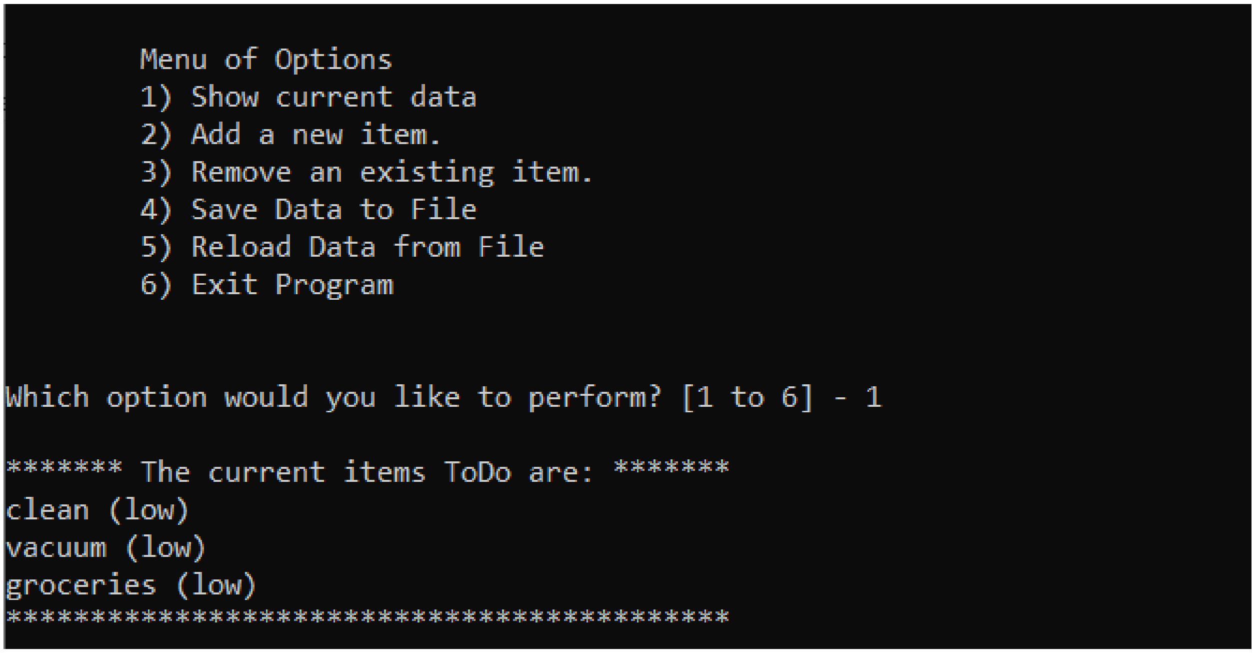


```
        FileProcessor.AddNewItem(strTask, strPriority, lstTable)
        IO.ShowCurrentItemsInList(lstTable)  # Show current data in the list/table
        continue  # to show the menu

    # Step 3.3 - Remove a new item to the list/Table
    elif(strChoice == '3'):

        # Step 3.3.a - Ask user for item and prepare searching while loop
        strKeyToRemove = input("Which TASK would you like removed? - ")  # get task
user wants deleted
        blnItemRemoved = False  # Create a boolean Flag for loop

        # Original code not replaced with function
        intRowNumber = 0  # Create a counter to identify the current dictionary row
in the loop

        # Step 3.3.b - Search though the table or rows for a match to the user's
input
        while(intRowNumber < len(lstTable)):
            if(strKeyToRemove ==
str(list(dict(lstTable[intRowNumber]).values())[0])):  # Search current row column 0
                del lstTable[intRowNumber]  # Delete the row if a match is found
                blnItemRemoved = True  # Set the flag so the loop stops
            intRowNumber += 1  # Increase counter to get next row

        # Step 3.3.c - Update user on the status of the search
        if(blnItemRemoved == True):
            print("The task was removed.")
        else:
            print("I'm sorry, but I could not find that task.")
        print()  # Add an extra line for looks

        #Step 3.3.d - Show the current items in the table
        IO.ShowCurrentItemsInList(lstTable)  # Show current data in the list/table
        continue  # to show the menu

    # Step 3.4 - Save tasks to the ToDoFile.txt file
    elif(strChoice == '4'):

        #Step 3.4.a - Show the current items in the table
        IO.ShowCurrentItemsInList(lstTable)  # Show current data in the list/table

        #Step 3.4.b - Ask if user if they want save that data
        if("y" == str(input("Save this data to file? (y/n) - ")).strip().lower()):  #
Double-check with user

            # Changed original code to function
            FileProcessor.WriteListDataToFile(strFileName, lstTable)

            input("Data saved to file! Press the [Enter] key to return to menu.")
        else:  # Let the user know the data was not saved
            input("New data was NOT Saved, but previous data still exists! Press the
[Enter] key to return to menu.")
        continue  # to show the menu
```

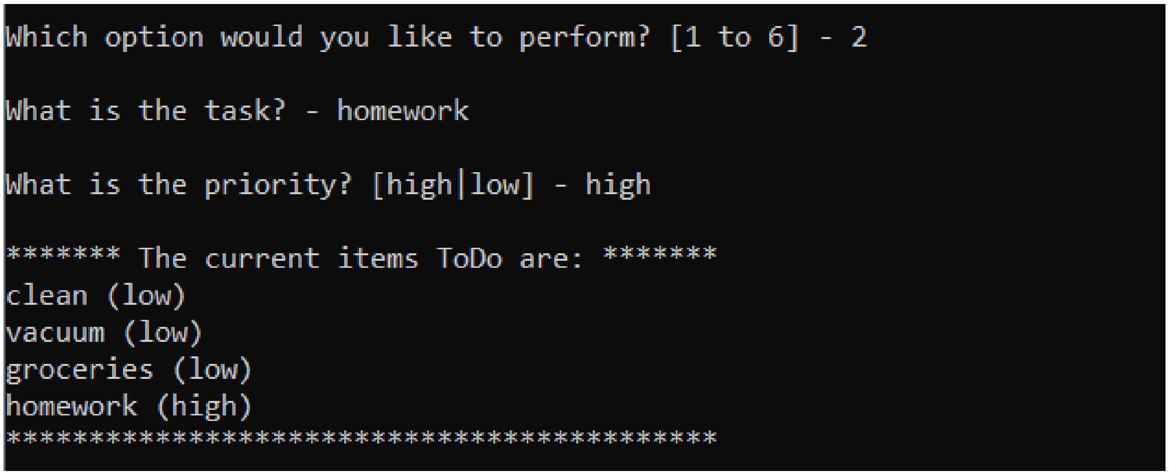


```
    # Step 3.5 - Reload data from the ToDoFile.txt file (clears the current data from the list/table)
    elif (strChoice == '5'):
        print("Warning: This will replace all unsaved changes. Data loss may occur!")  # Warn user of data loss
        strYesOrNo = input("Reload file data without saving? [y/n] - ")  # Double-check with user
        if (strYesOrNo.lower() == 'y'):
            lstTable.clear()  # Added to fix bug 1.1.2030
            FileProcessor.ReadFileDataToList(strFileName, lstTable)  # Replace the current list data with file data
            IO.ShowCurrentItemsInList(lstTable)  # Show current data in the list/table
        else:
            input("File data was NOT reloaded! Press the [Enter] key to return to menu.")
            IO.ShowCurrentItemsInList(lstTable)  # Show current data in the list/table
        continue  # to show the menu

    # Step 3.6 - Exit the program
    elif (strChoice == '6'):
        break   # and Exit
```

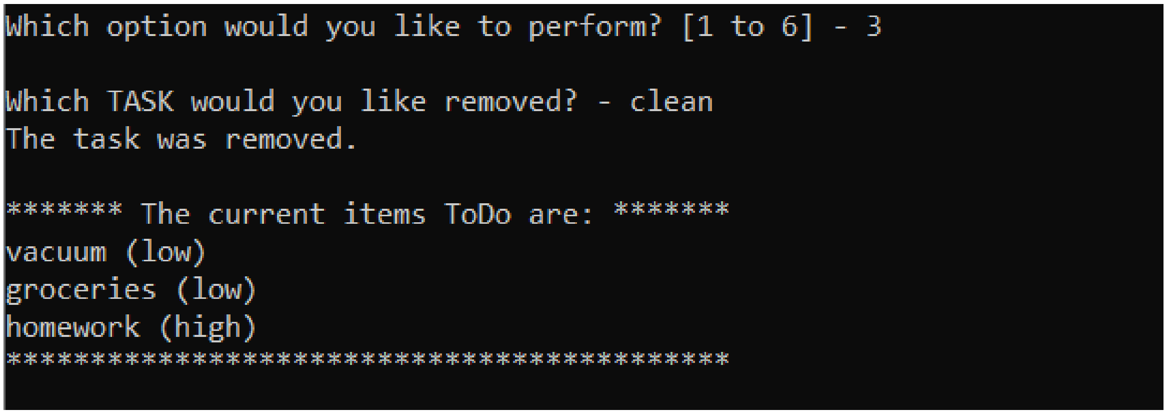
Double-clicking Assignment06.py directly from its file location prompts the script to initiate in a Python shell. Below is the final output after the user inputs an item and its value.



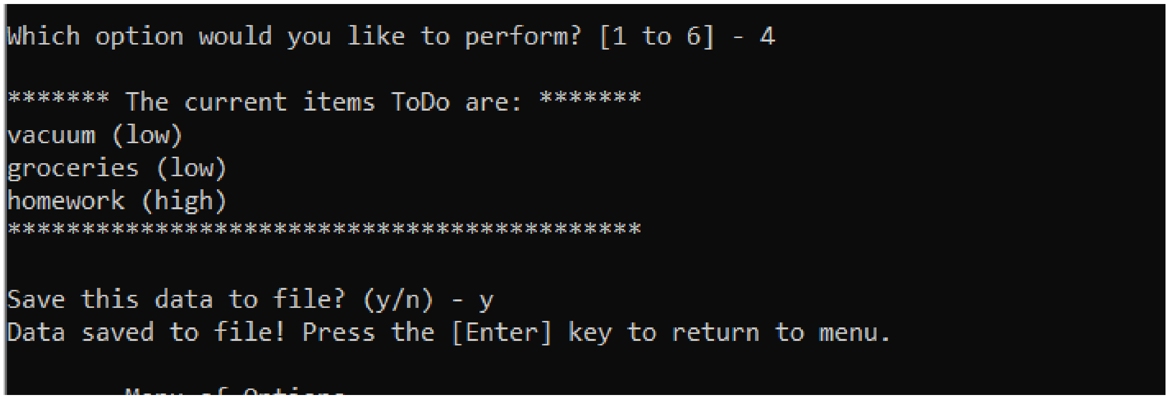
```
        Menu of Options
        1) Show current data
        2) Add a new item.
        3) Remove an existing item.
        4) Save Data to File
        5) Reload Data from File
        6) Exit Program


Which option would you like to perform? [1 to 6] - 1

******* The current items ToDo are: *******
clean (low)
vacuum (low)
groceries (low)
*******************************************
```



```
Which option would you like to perform? [1 to 6] - 2

What is the task? - homework

What is the priority? [high|low] - high

******* The current items ToDo are: *******
clean (low)
vacuum (low)
groceries (low)
homework (high)
*******************************************
```

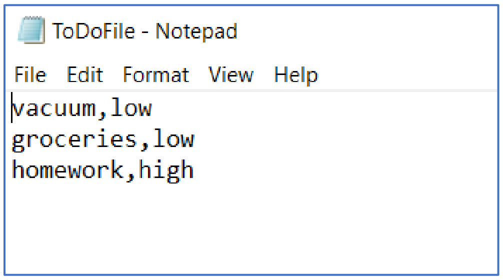


```
Which option would you like to perform? [1 to 6] - 3

Which TASK would you like removed? - clean
The task was removed.

******* The current items ToDo are: *******
vacuum (low)
groceries (low)
homework (high)
*******************************************
```



```
Which option would you like to perform? [1 to 6] - 4

******* The current items ToDo are: *******
vacuum (low)
groceries (low)
homework (high)
*******************************************


Save this data to file? (y/n) - y
Data saved to file! Press the [Enter] key to return to menu.

        Menu of Options
```

Below is the output in the newly created text file ToDoList.txt.



```
vacuum,low
groceries,low
homework,high
```