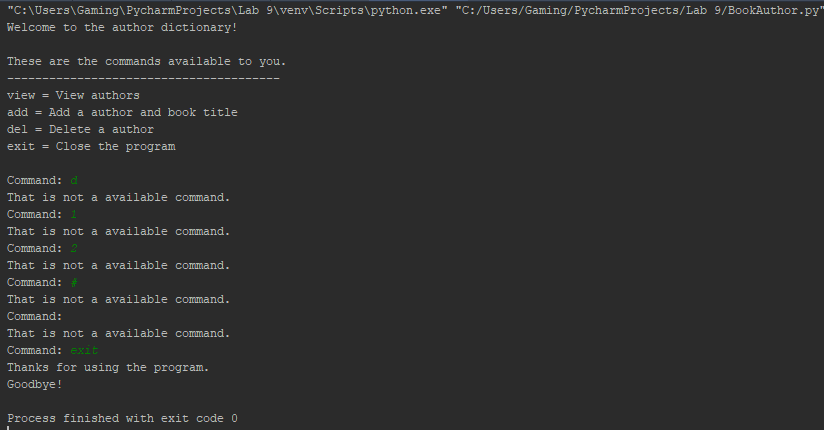
1. Problem Definition: Write a program that stores the names of your favorite authors and book titles in a dictionary as key-value pairs
2. Problem Analysis: Display a menu that lets the user view all the authors and titles, add a new author and book title, look up an author’s book title, change an existing book title, and delete an existing author and book title.
3. Program Algorithm:

* Give user a startup message that allows them to know what commands are available.
* Create a constant loop that will reactivate after the command.
* Give an input variable that is cycled through a loop to determine what the user wants to access.
* Give 4 options (View, Add, Delete, Exit)
* Each option will have a special function.

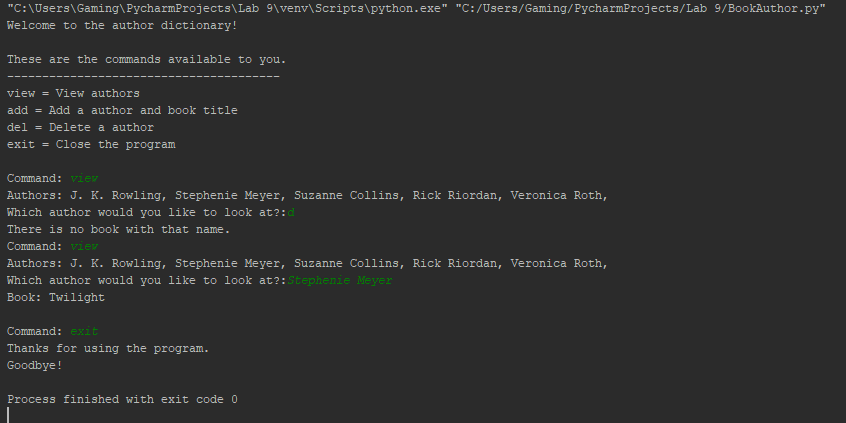
1. Program Code and Test:

# Author == "Darren Isaacson"  
# This program is designed to house a dictionary of books and author's names  
  
def main ():  
 authorandBooks = {"J. K. Rowling" : "Harry Potter", "Stephenie Meyer":"Twilight", "Suzanne Collins":"Hunger Games",  
 "Rick Riordan":"Percy Jackson", "Veronica Roth":"Divergent"} # Startup list  
 startupMessage() # Lets the user know what commands are available  
 try:  
 while True: # Constant loop  
 userInput = input("Command: ") # Command input  
 userInput = userInput.lower() # Changes input into lowercase  
   
 # This loop cycles through the users input and sends them to their selected function  
 if userInput == "view":  
 viewDict(authorandBooks) # View function  
 elif userInput == 'add':  
 addDict(authorandBooks) # Add function  
 elif userInput == 'del':  
 delDict(authorandBooks) # Delete function  
 elif userInput == 'exit': # Close program  
 print("Thanks for using the program.")  
 print("Goodbye!")  
 break  
 else:  
 print("That is not a available command.")  
 except:  
 print("There was an error in your program.")  
  
def startupMessage():  
 print("Welcome to the author dictionary! \n")  
 print("These are the commands available to you. \n---------------------------------------")  
 print("view = View authors")  
 print("add = Add a author and book title")  
 print("del = Delete a author")  
 print("exit = Close the program")  
 print()  
  
  
def viewDict(authorandBooks):  
 displayavaList(authorandBooks) # List display function  
 author = input("Which author would you like to look at?:")  
 author = author.title()  
 if author in authorandBooks:  
 book = authorandBooks[author] # This variable takes the input value and searches the list.  
 # If it exists it displays value.  
 print("Book: " + book + "\n")  
 else:  
 print("There is no book with that name.")  
  
def addDict(authorandBooks):  
 author = input("Enter in a author that you would like to add to your list.")  
 author = author.title()  
 if author in authorandBooks:  
 book = authorandBooks[author] # This variable takes the input value and searches the list for existing.  
 print("That author already exist in your list. Please try again.") # If it exists then it will not add another.  
 else:  
 book = input("Enter in the book for this author:") # If it doesn't exist then it ask to input the book added.  
 authorandBooks[author] = book # This assigns the book entered to the key input  
 print(book, "was added to your list. \n")  
  
def delDict(authorandBooks):  
 author = input("Which author in the list would you like to delete?:")  
 author = author.title()  
 if author in authorandBooks:  
 book = authorandBooks.pop(author) # This variable takes the input value and searches the list for existing.  
 print(book, "was deleted from your list. \n")  
 else:  
 print("That author does not exist.")  
  
  
def displayavaList(authorandBooks):  
 authors = list(authorandBooks.keys()) # Creates a key list and puts it under a variable  
 authors.sort # Takes the list and sorts it a-z  
 line = "Authors: "  
 for num in authors:  
 line += num + ", " # Iterates a space and comma for every author in the list  
 print(line)  
  
main()

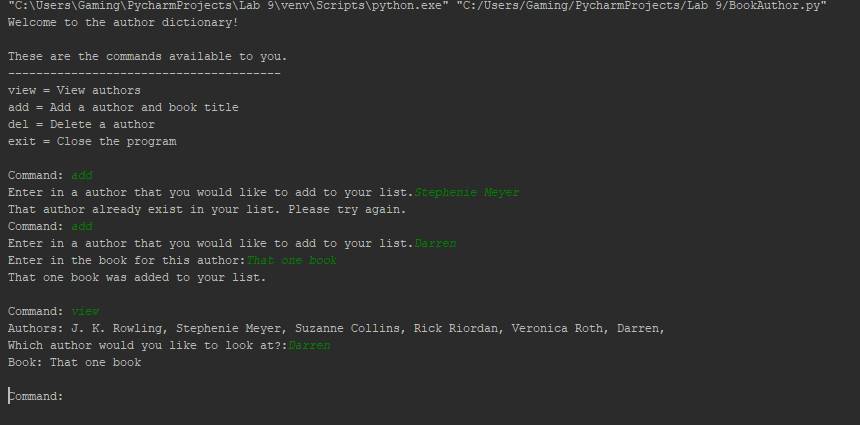
**TEST: Input**



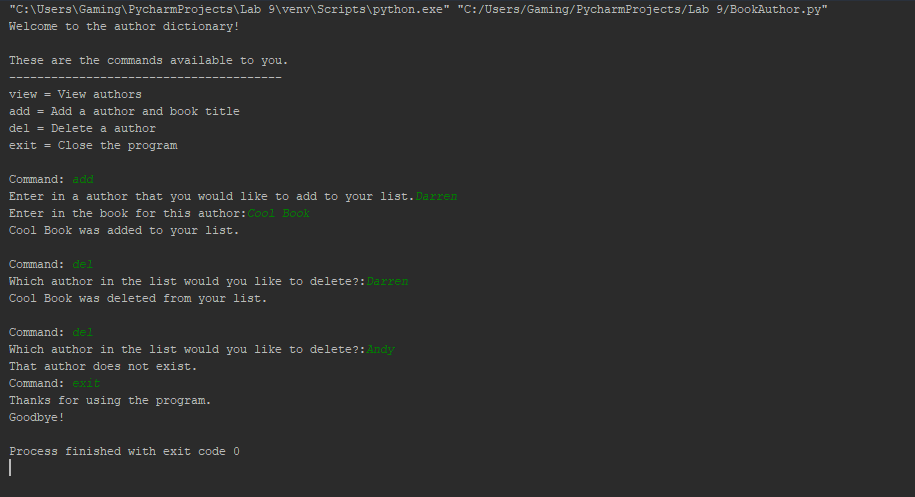
**View TEST**



**ADD Test**



**Delete Test**



**Input Correction Test**

