```
#-----Statement of Authorship----------------------------------------#
#
# This is an individual assessment task for QUT's teaching unit
# IFB104, "Building IT Systems", Semester 2, 2024. By submitting
# this code I agree that it represents my own work. I am aware of
# the University rule that a student must not act in a manner
# which constitutes academic dishonesty as stated and explained
# in QUT's Manual of Policies and Procedures, Section C/5.3
# "Academic Integrity" and Section E/2.1 "Student Code of Conduct".
#
# Put your student number here as an integer and your name as a
# character string:
#
student_number = 11912839
student_name = "Phuc Lam Vo"
#
# NB: All files submitted for this assessable task will be subjected
# to automated plagiarism analysis using a tool such as the Measure
# of Software Similarity (http://theory.stanford.edu/~aiken/moss/).
#
#--------------------------------------------------------------------#

#-----Assessment Task 1 Description----------------------------------#
#
# This assessment task tests your skills at processing large data
# sets, creating reusable code and following instructions to retrieve
# information. The incomplete Python program below is
# missing a crucial function. You are required to complete this
# function so that when the program runs it interacts with the user
# to perform a range of functionalities.
#  See the instructions in part A and part B for full details.
#
# Note that this assessable assignment is in multiple parts,
# simulating incremental release of instructions by a paying
# "client". This single template file will be used for all parts
# and you will submit your final solution as this single Python 3
# file only, whether or not you complete all requirements for the
# assignment.
#
# This file relies on one other Python modules but all of your code
# must appear in this file only. You may not change any of the code
# in the other module and you should not submit the other module
# with your solution. The markers will use their own copies of the
# other module to test your code, so your solution will not work
# if it relies on changes made to any other files.
#
#--------------------------------------------------------------------#

#-----Preamble-------------------------------------------------------#
#
# This section imports necessary functions used to execute your code.
# You must NOT change any of the code in this section, and you may
# NOT import any non-standard Python modules that need to be
# downloaded and installed separately

# Import standard Python modules needed to complete this assignment.
# You should not need to use any other modules for your solution.
# In particular, your solution must NOT rely on any non-standard
# Python modules that need to be downloaded and installed separately,
```

```
# because the markers will not have access to such modules.
from sys import exit as abort


# Confirm that the student has declared their authorship
if not isinstance(student_number, int):
      print('\nUnable to run: No student number supplied',
      '(must be an integer), aborting!\n')
      abort()
if not isinstance(student_name, str):
      print('\nUnable to run: No student name supplied',
      '(must be a character string), aborting!\n')
      abort()



#-----Student's Solution--------------------------------------------#
#
# Complete the assignment by replacing the dummy function below with
# your own function and any other functions needed to support it.
# All of your solution code must appear in this section. Do NOT put
# any of your code in any other sections and do NOT change any of
# the provided code except as allowed by the comments in the next
# section.
#

# All of your code goes in, or is called from, this function.
def title_length(booklist): #Get the length of the title
      average_length = 0
      return average_length

def search_word(booklist, word): #Searching for word in the booklist
      matching_title = "empty"
      return matching_title

def word_occurences(booklist, word): #Count how many times the word appear in the
booklist
      word_count = 0
      return word_count

def interact():
      user_booklist = input("Please enter your booklist:")
      #Ask for user booklist, if the length of the booklist is less than 50 then
user has to type again
      if len(user_booklist) < 50:
            user_booklist = input("Your booklist is too short, please enter at
least 50 characters.")
      # The user choose the action they want to do with the booklist
      print("Choose an action:")
      print("1.Get the average length of the title of the books")
      print("2.Search for a word in subtitles")
      print("3.Count word occurrences in titles and subtitles")
      user_choice = int(input("Enter your choice (1-3):"))
      if user_choice == 1:
            print("Average title length:",title_length(user_booklist))
      elif user_choice == 2:
            user_word = input("Enter a word to search:")
            print("Matching titles:",search_word(user_booklist,user_word))
```

```
      else:
            user_input_word = input("Enter a word to count occurrences:")
            print("Word
occurrences:",word_occurences(user_booklist,user_input_word))
      print("Goodbye!")

#-----Main Program to Run Student's Solution-------------------------#
#You must NOT change any of the code in this section.
if __name__ == "__main__":
    interact()
```