Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

Assessment type (☑):

☐ Questioning (Oral/Written)
☐ Practical Demonstration
☐ 3rd Party Report
☒ Other – Lab

Assessment Resources:

The base requirements this assessment task include:

- IDE or editor for developing Python programs *(only IDLE and PyCharm supported by the college)*
- Access to Office 365 & Microsoft Word
- Virtual machine

You may not need all these for every part in this assessment

Assessment Due :

This assessment is due after the weekly session, **Week 7, Friday 17:00**.

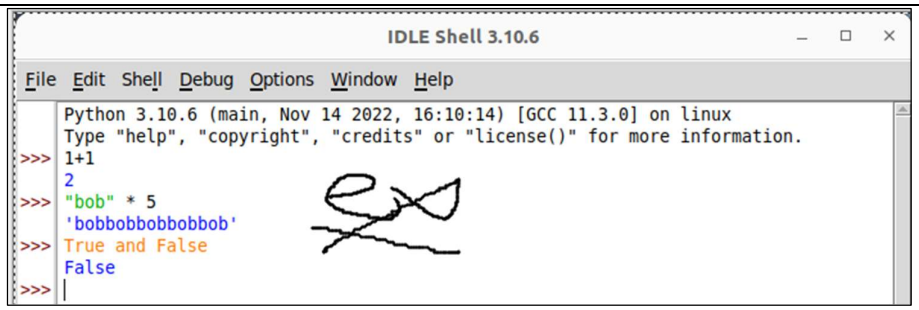
Assessment Instructions:

1. Your code must be written in IDLE or PyCharm IDEs. If you are using a different IDEs or a different structure for your application, then assistance from your lecturers may be limited (at best). Discuss with your lecturer before straying too far off the path!
2. All resources used should be referenced with the question. Answers may not be copied and pasted from any resource. All answers must be reworded to display your understanding.
3. You may only use Python functionality, methods and libraries which were taught in this unit.
4. First line of code in a program should have the student's name and number, as proof of authenticity.
5. Screenshots of all programs must be included **in** this document, with the appropriate question.
6. Screenshots of testing, showing your code works as intended, should be included with the relevant question.
7. Python programs should be named: XXX_Lab##_SYY_QZZ
   Replace XXX with your initials
   Replace ## with Lab number
   Replace YY with Section number,
   Replace ZZ with Question number
8. It is a submission requirement that all screen shots be signed in some way. Some acceptable examples of signed screen shots are shown below.

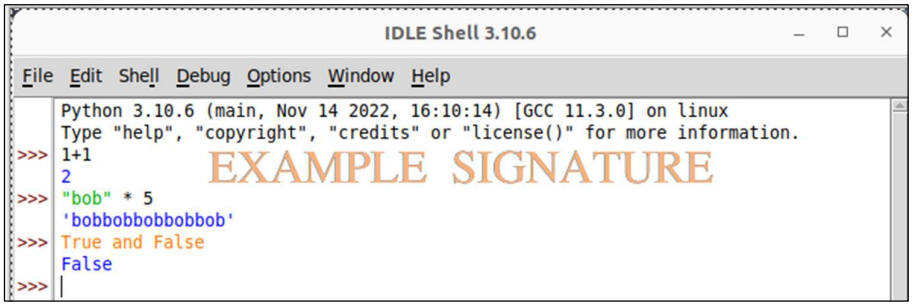
Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

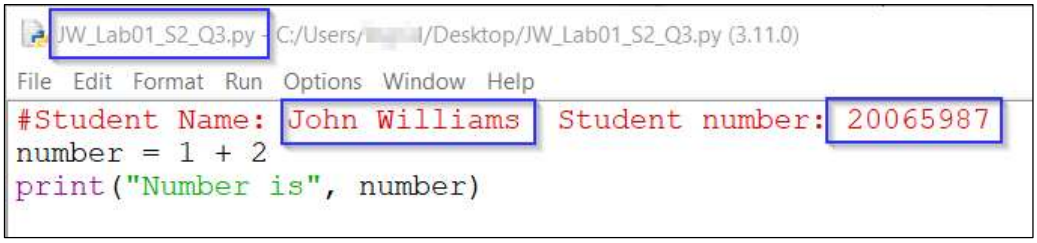
Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**



Example 1: Signed using a simple drawing tool.



Example 2: Water marked signature.



Example 3: Program named as prescribed, as well as first line comment with student name and number. Program saved as pre-described.

9. All python programs must be included in the submission, as well as this document.

Assessment Instrument:

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

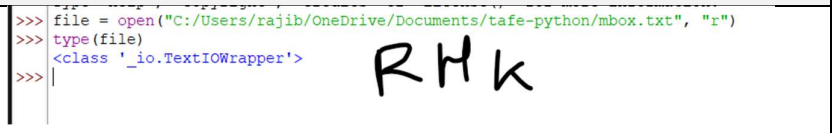
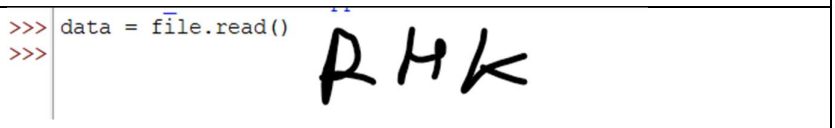
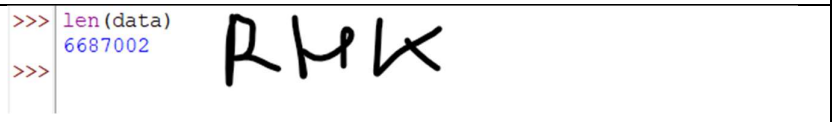
Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

Section 1: Opening files

Opening and working with the contents of files will be a crucial skill for the upcoming project. Run the following exercises in the Python shell/interpreter.

- Start by downloading the sample text file from the P4e web site (mbox.txt).
- Open the mbox.txt file using code like the following (ensure you modify the file path to match your location of **mbox.txt**, in this example, the file was stored in the /tmp/ directory on a Linux machine):

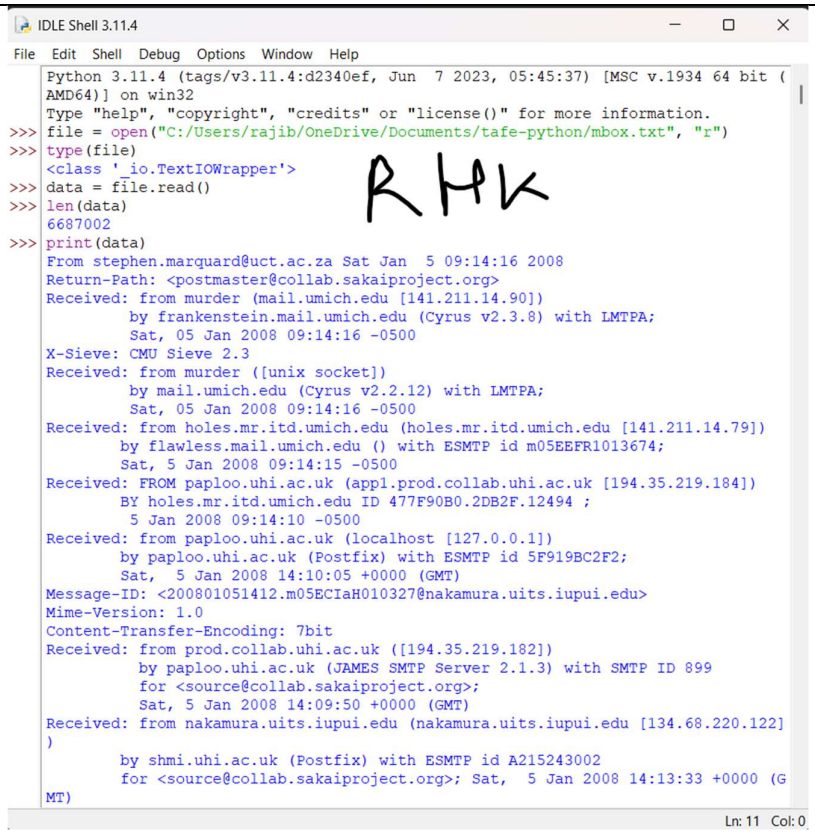
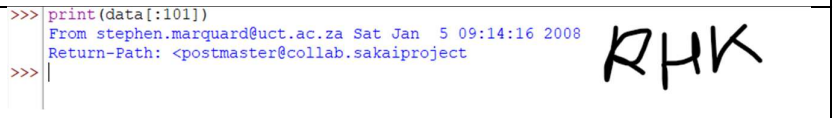
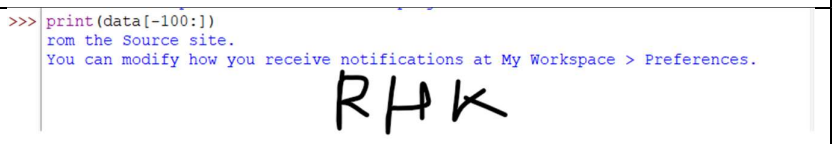
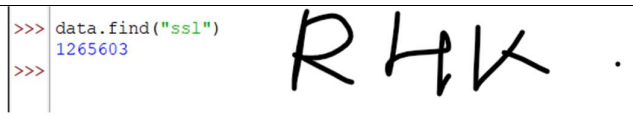
```
>>> file = open("/tmp/mbox.txt","r")
```

Question	Answer / Screenshot
1. Using the **type()** function to determine what type of object **file** is.	
2. Use the **read()** function save the contents of the file to a new variable.	
3. Use **len()** to calculate how many characters in total are in the variable's contents.	

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

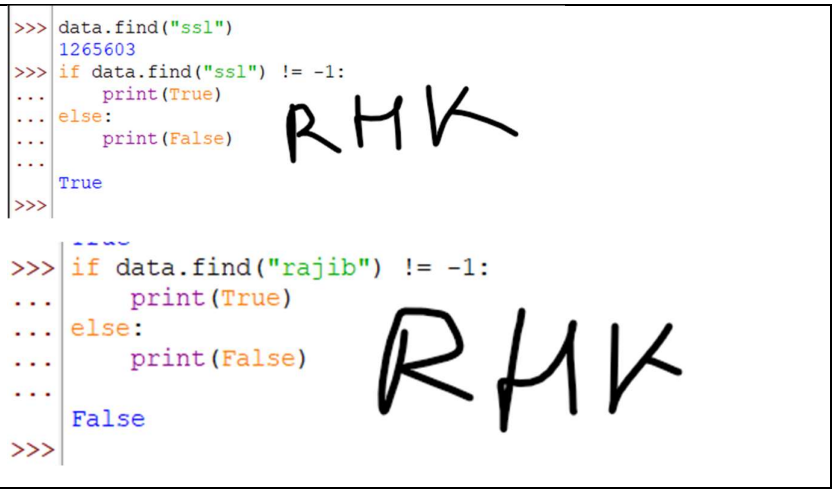
Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

4. Print the new variable (it should list everything in the file you read). If you get the following result, double click on it to expand Squeezed text (140544 lines).	
5. Print only the first 100 characters in the variable.	
6. Print only the last 100 characters in the variable.	
7. Check if the word **ssl** appears in the file list. (The answer should return True).	
8. Create a simple if statement to check if any word you want appears in the list. The if	

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

statement should produce True or False based on the word (demonstrate this with two different words to show your statement works as intended).	

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

Section 2: Create and modify your own files

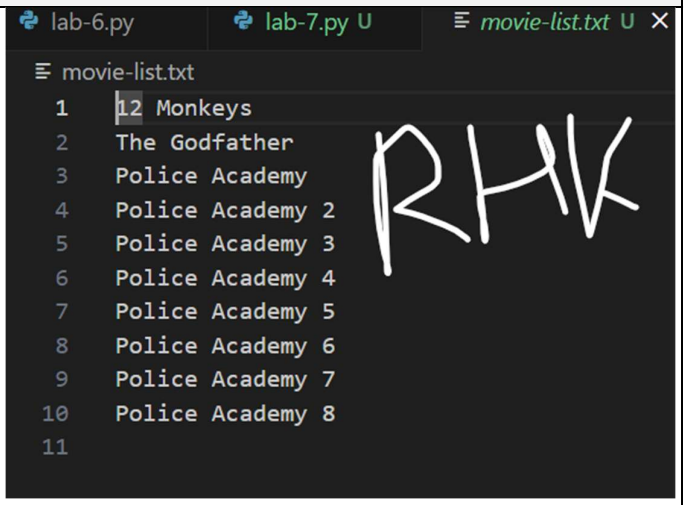
You will now begin some practice loading and writing to files using Python.

Step 1:

- In the Python interpreter, create ten new variables with movie titles in them; you should name them after your favourite movies. Example:

```
>>> movie01 = "12 Monkeys"
>>> movie02 = "The Godfather"
>>> movie03 = "Police Academy"
```

- Use open() to open a new file in **write** mode.
- Use write() to write your data to the file.
- Use close() to complete the write process and release the file.
- Double check the contents of the file using your operating system to ensure the data was correctly written to the file.

Code	File Content
	

Step 2:

- Create five more variables with five more movie titles that you enjoy allocated to the variables.
- *Again*, **open()** the file in write mode and **write()** your movies to the file and **close()** it.

Code	File Content

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

```
movie11 = "Police Academy 9"
movie12 = "Police Academy 10"
movie13 = "Police Academy 11"
movie14 = "Police Academy 12"
movie15 = "Police Academy 13"
with open("movie-list.txt", "w") as f:
    f.write(movie11 + "\n")
    f.write(movie12 + "\n")
    f.write(movie13 + "\n")
    f.write(movie14 + "\n")
    f.write(movie15 + "\n")
    f.close()
```

RHK

lab-6.py | lab-7.py U | movie-list.txt U

movie-list.txt

```
1  Police Academy 9
2  Police Academy 10
3  Police Academy 11
4  Police Academy 12
5  Police Academy 13
6
```

RHK

Question	**Answer**
Examine the file in your operating system. 1) What happened to the file? What is now inside it?	The old content replaced with the new content
2) Conclusion: What does the write function do?	Write function is writing the specified content in the file

Step 3

- Create five more variables with movie titles to use
- Use **open()** to open the file in **append** mode and **write()** the data to it, **close()** it afterwards.

Code

```
movie16 = "Police Academy 14"
movie17 = "Police Academy 15"
movie18 = "Police Academy 16"
movie19 = "Police Academy 17"
movie20 = "Police Academy 18"
with open("movie-list.txt", "a") as f:
    f.write(movie16 + "\n")
    f.write(movie17 + "\n")
    f.write(movie18 + "\n")
    f.write(movie19 + "\n")
    f.write(movie20 + "\n")
    f.close()
```

RHK

File Content

lab-6.py | lab-7.py U | movie-list.txt U

movie-list.txt

```
1  Police Academy 9
2  Police Academy 10
3  Police Academy 11
4  Police Academy 12
5  Police Academy 13
6  Police Academy 14
7  Police Academy 15
8  Police Academy 16
9  Police Academy 17
10 Police Academy 18
11
```

RHK

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

Question	Answer
Examine the file in your operating system. 1) What happened to the file? What is now inside it?	Now the code appended the new content to the existing content
2) What is the difference between *write* and *append* when using **open()**?	Write overwrites the existing content and append adds the new content without overwriting it
3) If you do not parse any mode (write, append, read etc) to the **open()** function, what mode does it use by default?	read

Step 4

- Lastly, try to **open()** your text file in read mode but do not **close()** it yet.
- In your operating system, try to delete the file Python currently has open. What happens?

```
53  import os
54  newFile = open("movie-list.txt", "r")
55  os.remove("C:/Users/rajib/OneDrive/Documents/tafe-python/movie-list.txt")
56

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  POLYGLOT NOTEBOOK   Python

PS C:\Users\rajib\OneDrive\Documents\tafe-python> & C:/Users/rajib/AppData/Local/Programs/Python/Python311/python.exe c:/U
sers/rajib/OneDrive/Documents/tafe-python/lab-7.py
Traceback (most recent call last):
  File "c:\Users\rajib\OneDrive\Documents\tafe-python\lab-7.py", line 55, in <module>
    os.remove("C:/Users/rajib/OneDrive/Documents/tafe-python/movie-list.txt")
PermissionError: [WinError 32] The process cannot access the file because it is being used by another process: 'C:/Users/r
ajib/OneDrive/Documents/tafe-python/movie-list.txt'
PS C:\Users\rajib\OneDrive\Documents\tafe-python>
```

RHK

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

Section 3: A practical example on a server

This section of the lab is optional, it is not required to be marked as satisfactory for this assessment.

The process of working with (opening, writing, and closing files) can vary depending on the operating system you are working with. This section will ask you to demonstrate writing a script that will run and work on a Linux server system.

The system used as a demonstration in this example is **Rocky Linux 9 minimal**. You can obtain this VM from BlackBoard or build your own.

Screen shots are provided as an example of what you should see, ensure you provide your own signed screen shots to demonstrate that you have completed this section in full.

1. Start the VM.
2. Log in with:
    a. Username **user**
    b. Password **Password1**
3. Verify your virtual machine has internet connectivity by using the command **curl http://example.com**. If this command was successful, you should see the html code used to build the example.com web site.

```
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
```

4. Install **Python** and **nano** using the command **sudo dnf install nano python3** (they may already be installed depending on your circumstances).
5. Determine which version of Python you have installed by using the command **python -V**.
6. Figure out where the Python interpreter is located on the file system by using the command **which python**.
7. Open a new file (**example.py)** by using the command **nano example.py**.

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

8. Write a script to show the first 10 characters in any file you parse to it using command line arguments:

```
 GNU nano 5.6.1
#!/usr/bin/python

# Import the sys class for arguement passing
import sys

# Grab the name of the file to check
fileArg = sys.argv[1]

# Open the file
file = open(fileArg)

# Read the file
contents = file.read()

# Close the file
file.close()

# Print the first 10 of file
print(contents[:10])
```

Note, the first line in any bash (Linux interpreter) script is known as the **shebang** and must provide the path to the Python interpreter as found in an earlier step. This tells the Linux bash know which interpreter it should run the script though.

9. Save and close with **Control + X**.
10. When done, make the script executable with the command **chmod +x /example.py**.
11. Run the script several times providing any file as an argument (example files may be found in /etc/).

```
[demo-user@demo-rocky ~]$ ./example.py /etc/rsyslog.conf
# rsyslog
[demo-user@demo-rocky ~]$ ./example.py /etc/krb5.conf
# To opt o
[demo-user@demo-rocky ~]$ ./example.py /etc/passwd
root:x:0:0
```

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

Section 4: A challenge

Complete the following challenge by creating code and demonstrating its functionality.
You are required to create a script on any operating system (Linux/Windows etc) that can

1) open a file and
2) count all occurrences of a select letter and
3) check if a certain word occurs in the file.
4) Print the results to the screen
5) The script must then save those results to a file named **results**.

Your script must ask the user for the following information:

- The name of a file to open.
  - If the file cannot be found, then keep prompting until a valid file name is entered
- Which letter to count the occurrences of, in the file
  - If a single letter of the alphabet is not entered, then keep prompting until entered
- A string used to check for the existence of in the file.

Task:

1. Write pseudocode to demonstrate how your script will function.
2. Once your pseudocode has been created, complete your solution by translating your pseudocode into Python code.

Note: To be marked as satisfactory for this task, your solution must:

- Produce and submit your pseudocode.
- Demonstrate through screen shots, that your solution addresses all the above points.
- You must submit the code file with this document
- The code file must contain comprehensive commenting
- The output must be formatted as shown below

```
Enter the name of the file: C:\TAFE\example.txt
C:\TAFE\example.txt does not exist. Please try again.
Enter the name of the file: C:\TAFE\text.txt
Enter a letter to count: ty
Only a single letter of the alphabet is allowed. Please try again.
Enter a letter to count: 6
Only a single letter of the alphabet is allowed. Please try again.
Enter a letter to count: t
Enter a string to search for: s: T
There are 13 ocurrences of the letter "t".
The string "s: T" exists in the file.
```

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

Pseudocode:

1.
```
# get a correct file name from the user until the file is found
# get a letter and a string from the user once the file is found
# the leter should be a single letter
# read the file and get the no of occurences of the letter
# check if the string exists in the file
# print out the result
# write the result in a new file      You, 1 second ago • Uncommitted changes
```
RHK

Python Code

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**

```python
# initiating the variables required to accomplish the task
fileName = None
file = None
letter = None
letterCount = 0

# promting for the filename until a file can be found
while not fileName or not file:
    fileName = input("Enter the name of the file: ")
    # trying to open the file name if the user provided value is not empty
    if fileName.strip():
        try:
            # opening the file
            file = open(fileName.strip())
        # handling file not found exception
        except FileNotFoundError:
            print(f"{fileName} does not exist. Please try again.")
    # asking for the letter & string user wants to search once the file is found
    if file != None:
        # promting user to enter a valid letter
        while not letter:
            searchLetter = input("Enter a letter to count: ")
            # checking the validity of user input
            if searchLetter.strip().isalpha() and len(searchLetter) == 1:
                letter = searchLetter.strip()
            else:
                print(
                    "Only a single letter of the alphabet is allowed. Please try again.")
        # resding the file
        data = file.read()
        # couting the occurrences of the letter
        letterCount = data.count(letter)
        # asking for the string user wants to search
        userString = input("Enter a string to search for: ")
        # printing the count of the letter
        result1 = f'There are {letterCount} occurrences of the letter "{letter}".'
        result2 = ""
        print(result1)
        # searching for the string and printing the result
        if data.find(userString) != -1:
            result2 = f'The string "{userString}" exists in the file.'
            print(result2)
        else:
            result2 = f'The string "{userString}" does not exist in the file.'
            print(result2)
        with open("result.txt", "a") as f:
            f.write(result1 + "\n")
            f.write(result2 + "\n")      You, 1 minute ago • Uncommitted changes
```

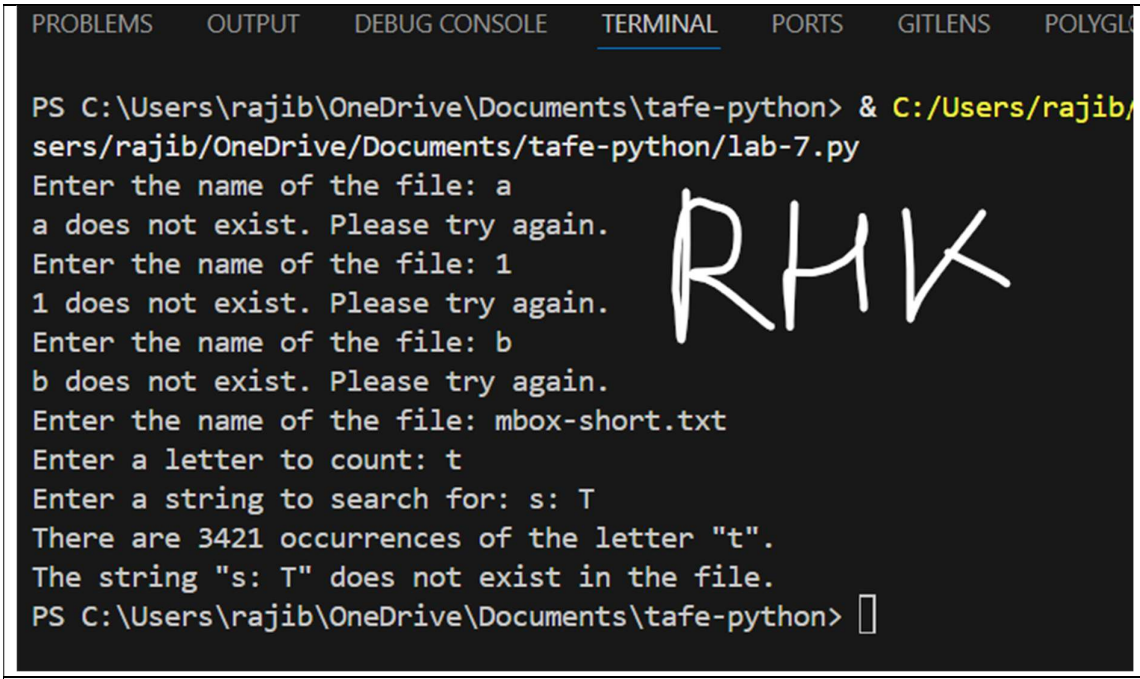
RHK

Program screenshots showing all possible outcomes.

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**



Screenshots of:

- Input file and
- Result file output

Government of Western Australia
North Metropolitan TAFE

Assessment Task: Lab 7

Qualification national code and title	22603VIC Certificate IV in Cybersecurity
Unit/s national code/s and title/s	**ICTPRG434 - Automate processes** **ICTPRG435 - Write script for software applications**