1811ICT/2807ICT/7001ICT Programming Principles Workshop 7

School of Information and Communication Technology

Griffith University

|  |  |
| --- | --- |
| *Goals* | This workshop focusses on everything in the course up to files. |
| When | Week 8 |
| Marks | 3 |
| Due | Pre-workshop questions before the start of the above mentioned workshops  Workshop programming problems by 11:59pm on 9 May |

# Before your workshop class:

* Read all of this document.
* Review the lecture notes sections 1 to 20.
* **Complete the pre-workshop questions (1 mark) posted on the course website and submit the answers for marking**.

# Workshop activities

At any stage, when you are stuck, *ask your tutor*!

## Problem 1

*Problem:* Write a program that prompts for the names of a source file to read and a target file to write, and copy the content of the source file to the target file, but with all empty lines removed, then output the number of empty lines removed.

|  |
| --- |
| Source file name: string\_doc.txt  Target file name: string\_doc\_nonempty.txt  Lines removed: 16 |

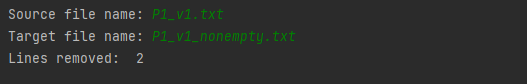
*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following two scenarios:

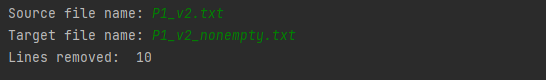
* Use the attached file P1\_v1.txt as the source file to read. Use P1\_v1\_nonempty.txt as the target file name.
* Use the attached file P1\_v2.txt as the source file to read. Use P1\_v2\_nonempty.txt as the target file name.

***Copy your code here***

source\_file = input("Source file name: ")  
target\_file = input("Target file name: ")  
  
with open(source\_file, "r") as f1, open(target\_file, "w") as f2:  
 lines = f1.readlines()  
 empty\_lines = 0  
 for line in lines:  
 if line.strip() != "":  
 f2.write(line)  
 else:  
 empty\_lines += 1  
  
print("Lines removed: ", empty\_lines)

***Insert your screenshots here***





## Problem 2

*Problem:* Write a program that prompts for the name of a file, then prints the first two lines and the last two lines of the file.

|  |
| --- |
| File name: yesterday.txt  Output:  Yesterday Once More  When I was young  I would sing to then  And I’d memorize each... |

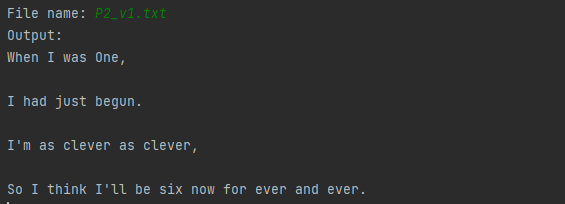
*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following two scenarios:

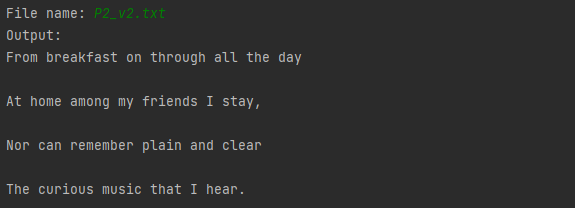
* Use the attached file P2\_v1.txt as the source file to read.
* Use the attached file P2\_v2.txt as the source file to read.

***Copy your code here***

filename = input("File name: ")  
  
with open(filename, "r") as f:  
 lines = f.readlines()  
 print("Output:")  
 print(lines[0])  
 print(lines[1])  
 print(lines[-2])  
 print(lines[-1])

***Insert your screenshots here***





## Problem 3

*Problem:* Write a program that prompts for the name of a file containing numbers in each line, prints the average of each line. Assume each line contains numbers only and they are separated by spaces.

File name: scores.txt

The average of line 1 is 60.0

The average of line 2 is 91.75

The average of line 3 is 48.75

The average of line 4 is 56.25

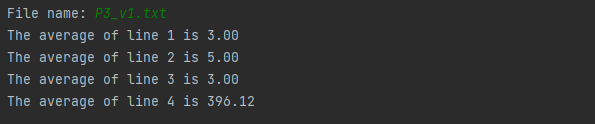
*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following two scenarios:

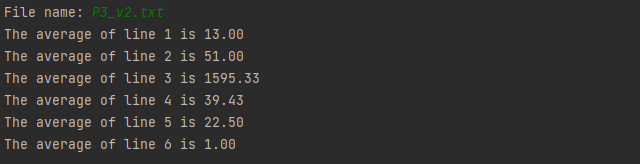
* Use the attached file P3\_v1.txt as the source file to read.
* Use the attached file P3\_v2.txt as the source file to read.

***Copy your code here***

filename = input("File name: ")  
  
with open(filename, "r") as f:  
 lines = f.readlines()  
 for i, line in enumerate(lines):  
 numbers = line.split()  
 total = sum(map(float, numbers))  
 average = total / len(numbers)  
 print(f"The average of line {i+1} is {average:.2f}")

***Insert your screenshots here***





## Problem 4

*Problem:* The Unix tool wc counts the numbers of characters, words and lines in a file. Write your own version of wc that prompts for the name of the file to read, then prints the counts. Assume a word may contain letters, digits, symbols and their mixture, but not space. Hyphenated words, e.g. large-scale, shall be considered as one word.

|  |
| --- |
| File name: python.txt  Characters: 1227  Words: 176  Lines: 10 |

*Answer*: Copy your code in the space given below and insert screenshots of your program output for the following two scenarios:

* Use the attached file P4\_v1.txt as the source file to read.
* Use the attached file P4\_v2.txt as the source file to read.

***Copy your code here***

filename = input("File name: ")  
  
with open(filename, "r") as f:  
 content = f.read()  
 num\_chars = len(content)  
 num\_words = len(content.split())  
 num\_lines = content.count("\n") + 1  
 print(f"Characters: {num\_chars}")  
 print(f"Words: {num\_words}")  
 print(f"Lines: {num\_lines}")

***Insert your screenshots here***





# Submission and marking

The pre-workshop can be accessed and submitted online using the provided link in the course website. Students get 1 mark if they get >50% in pre-workshop questions, or 0.5 mark if they get 0%-50% in pre-workshop questions, or 0 marks without any attempt.

For workshop tasks, please submit this document with copied codes and inserted screenshots using the provided submission link in the course website. Students get 2 marks if they complete three or more problems correctly, or 1 mark if they complete one or two problems correctly, or 0 marks without any attempt.