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 | Workshops from Friday 6 May to Thursday 12 May |
| Marks | 3 |
| Due | Pre-workshop questions before the start of the above mentioned workshops  Workshop programming problems by 11:59pm on 15 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***

# Matthew Prendergast

# 7th May, 2022 - Problem 1 (Workshop - Week 7)

# Prompt user to input a the file names.

file\_read = input("Source file name: ")

file\_write = input("Target file name: ")

# Initialise the count to 0.

count = 0

# Open a file to read and create a file to write to.

f\_read = open(file\_read, mode = 'r')

f\_write = open(file\_write, mode = 'w')

# Loop over the file and write all non-empty strings. Incerment count for all empty strings.

for line in f\_read:

    if line != "\n":

        f\_write.write(line)

    else:

        count += 1

# Print the result.

print("Lines removed:", count)

f\_read.close()

f\_write.close()

***Insert your screenshots here***

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

## 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***

# Matthew Prendergast

# 7th May, 2022 - Problem 2 (Workshop - Week 7)

# Prompt user to input a file name.

file\_name = input("File name: ")

# Open the file and read all lines into a list

fhand = open(file\_name, mode = "r")

ls = fhand.readlines()

# Print the result.

print("Output:")

print(ls[0], ls[1], ls[-2], ls[-1], sep="")

fhand.close()

***Insert your screenshots here***

Text

Description automatically generated

Text

Description automatically generated

## 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***

# Matthew Prendergast

# 7th May, 2022 - Problem 3 (Workshop - Week 7)

# Prompt user to input a file name.

file\_name = input("File name: ")

# Open the file and read all lines into a list

fhand = open(file\_name, mode = "r")

# Initialise the line count.

count = 1

# Loop through all of the lines in the file.

for line in fhand:

    # Create a new list for each line of scores. Cast to integers. Find the average.

    scores = line.strip().split(" ")

    for i in range(len(scores)):

        scores[i] = int(scores[i])

    total = sum(scores) / len(scores)

    # Print the result. Increment the line count.

    print("The average of line", count, "is", total)

    count += 1

fhand.close()

***Insert your screenshots here***

Text

Description automatically generated

Text

Description automatically generated

## 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***

# Matthew Prendergast

# 7th May, 2022 - Problem 3 (Workshop - Week 7)

# Prompt user to input a file name.

file\_name = input("File name: ")

# Open the file and read all lines into a list

fhand = open(file\_name, mode = "r")

# Initialise the variables.

characters = 0

words = 0

lines = 0

for line in fhand:

    # Count the characters.

    for i in line:

        characters += 1

    # Count the words.

    reader = line.strip().split(" ")

    words += len(reader)

    # Count the lines.

    lines += 1

print("Characters:", characters)

print("Words:", words)

print("Lines:", lines)

fhand.close()

***Insert your screenshots here***

Text

Description automatically generated

Text

Description automatically generated

# 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.