1811ICT/2807ICT/7001ICT Programming Principles Workshop 6

School of Information and Communication Technology

Griffith University

Ghadeer Alkhunaizi

|  |  |
| --- | --- |
| *Goals* | This workshop focusses on lists, indexing, slices, list methods, and/or tuples. |
| When | Week 7 |
| Marks | 2 |
| Due | Workshop programming problems by 11:59pm on 30 May |

# Before your workshop class:

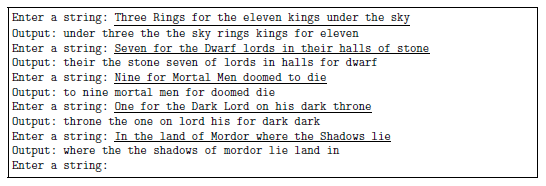
* Read all of this document.
* Review the lecture notes sections 1 to 17.
* **Complete the pre-workshop questions posted on the course website**.

# Workshop activities

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

## Problem 1

*Problem:* Write a program that reads strings (without digits or symbols in the string) typed by the user until an empty string is entered. For each string, convert all words to lowercase, then sort and print the words into descending order lexicographically. Hint: use split function to split a string into a list, then sort it with a method.



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

* Enter a string: The Old Woman Who Lived in a Shoe
* Enter a string: Birds of Prey and the Fantabulous Emancipation of One Harley Quinn

***Copy your code here***

*a = 0*

*while a == 0:*

*String = str(input("Enter a string:"))*

*lower = String.lower()*

*split = lower.split()*

*split.sort(reverse=True)*

*output = ("output:" + "".join(split))*

*print(output)*

***Insert your screenshots here***

A screen shot of a computer program

Description automatically generated

## Problem 2

*Problem:* Write a program that allows the user to enter two lists of integers, calculate the sum of the first and the last integers in each list, and print the larger sum. In the event of a tie, print Same. When there is only one integer in the list, the sum is the integer itself.

List 1: 1 2 3 4 5

List 2: 5 6 7

Output: 12

List 1: 4 3 10 1

List 2: 9

Output: 9

List 1: 4 3 2 1

List 2: 1 2 3 4

Output: Same

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

* List 1: [2, 4, 5, 7, 8, 9]; List 2: [6, 5]
* List 1: [7, 3, 23, 5, 12]; List 2: [11, 23, 4, 1, 3]

***Copy your code here***

*list1 = [int(x) for x in input("List 1: ").split()]*

*list2 = [int(x) for x in input("List 2: ").split()]*

*if len(list1) > 1:*

*sum1 = list1[0] + list1[-1]*

*else:*

*sum1 = list1[0]*

*if len(list2) > 1:*

*sum2 = list2[0] + list2[-1]*

*else:*

*sum2 = list2[0]*

*if sum1 > sum2:*

*print("output:",sum1)*

*elif sum2 > sum1:*

*print("output:",sum2)*

*else:*

*print("output: same")*

***Insert your screenshots here***

A screenshot of a computer program

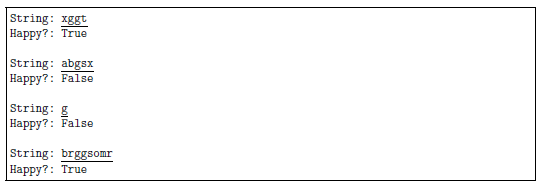
Description automatically generated

A screen shot of a computer program

Description automatically generated

## Problem 3

*Problem:* We’ll say that a lowercase ’g’ in a string is ”happy” if there is another ’g’ immediately to its left or right. Write a function to print True if all the g’s in the given string are happy, otherwise, print False.

**

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

* String: poggpoggpogg
* String: daggergoo

***Copy your code here***

*def are\_all\_gs\_happy(input\_string):*

*# Iterate through the string to check each 'g'*

*for i in range(len(input\_string)):*

*if input\_string[i] == 'g':*

*# Check if the 'g' is happy (has another 'g' adjacent)*

*if (i > 0 and input\_string[i - 1] == 'g') or (i < len(input\_string) - 1 and input\_string[i + 1] == 'g'):*

*continue # 'g' is happy, move to next character*

*else:*

*return False # 'g' is not happy*

*return True # All 'g's are happy*

*# Test cases*

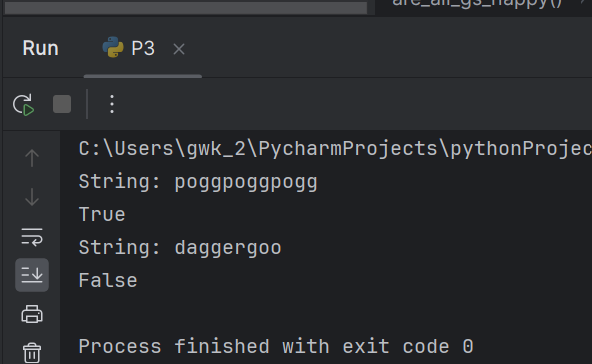
*strings = ["poggpoggpogg", "daggergoo"]*

*for s in strings:*

*print(f"String: {s}")*

*print(are\_all\_gs\_happy(s))*

***Insert your screenshots here***



## Problem 4

*Problem:* Write a program that allows the user to enter the marks of 5 students in 4 courses, and outputs the highest average marks for students and courses. Hint: Consider 2 dimensional lists, i.e. the element of a list is a list.

Student 1 (courses 1-4): 50 60 70 60

Student 2 (courses 1-4): 100 90 87 90

Student 3 (courses 1-4): 70 100 90 90

Student 4 (courses 1-4): 30 65 50 50

Student 5 (courses 1-4): 58 50 74 43

The highest average mark of students: 91.75

The highest average mark of courses: 74.2

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

* Student 1 (courses 1-4): 85 70 65 60

Student 2 (courses 1-4): 50 55 75 90

Student 3 (courses 1-4): 67 87 77 55

Student 4 (courses 1-4): 97 88 86 90

Student 5 (courses 1-4): 35 50 55 60

* Student 1 (courses 1-4): 58 77 68 58

Student 2 (courses 1-4): 49 67 89 78

Student 3 (courses 1-4): 56 86 76 66

Student 4 (courses 1-4): 58 88 83 78

Student 5 (courses 1-4): 76 69 77 80

***Copy your code here***

*def get\_average(list):*

*total = 0*

*for i in list:*

*total += 1*

*return total/len(list)*

*Student\_marks = []*

*for list in range(5):*

*print("Enter Marks for student",list+1,":",end="")*

*marks = [int(x) for x in input().split(" ")]*

*Student\_marks.append(marks)*

*average\_Student\_Marks =[]*

*for marks in Student\_marks:*

*avg = get\_average(marks)*

*average\_Student\_Marks.append(avg)*

*print("The highest average mark of students:",max(average\_Student\_Marks))*

*course1 =[]*

*course2 =[]*

*course3 =[]*

*course4 =[]*

*for marks in Student\_marks:*

*course1.append(marks[0])*

*course2.append(marks[1])*

*course3.append(marks[2])*

*course4.append(marks[3])*

*average\_course1 = get\_average(course1)*

*average\_course2 = get\_average(course2)*

*average\_course3 = get\_average(course3)*

*average\_course4 = get\_average(course4)*

*average\_course\_mark = [average\_course1,average\_course2,average\_course3,average\_course4]*

*print("The highest average mark of courses: ", max(average\_course\_mark))*

***Insert your screenshots here***

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Submission and marking

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.