2807/7001ICT Programming Principles (I), Trimester 3, 2018
Workshop 6

School of Information and Communication Technology
Griffith University

November 5, 2018

Module	2
When	Day 6
Goals	This workshop focusses on lists, indexing, slices, list methods, and/or tuples.
Marks	5
Due	Problem 1 by the end of this workshop; problems 2 to 4 by the *beginning* of the next workshop.

1 Preparation

Before your workshop class:

- Read all of this document.
- Review the lecture notes sections 1 to 17.
- Bring some paper (a print-out of this document is best) and writing implements.
- Bring a storage device, such as a portable hard drive and cable, or a USB drive.

2 Pre-workshop questions

There are no pre-workshop questions for this workshop.

3 Workshop activities

3.1 Marking last workshop's problems

If you have problems that still need marking from the previous workshop, get them marked at the *start* of this one.

3.2 Problem 1 (1 mark)

This problem is due by the end of this workshop.

> *Problem:* Write a program that reads strings typed by the user until an empty string in entered, then prints all of the strings sorted into ascending (actually, non-descending, lexicographic) order.

```
Enter a string: On a little piece of wood,
Enter a string: Mr. Spikky Sparrow stood;
Enter a string: Mrs. Sparrow sate close by,
Enter a string: A-making of an insect pie,
Enter a string: For her little children five,
Enter a string: In the nest and all alive,
Enter a string: Singing with a cheerful smile
Enter a string: To amuse them all the while,
Enter a string: Twikky wikky wikky wee,
Enter a string: Wikky bikky twikky tee,
Enter a string: Spikky bikky bee!
Enter a string:
A-making of an insect pie,
For her little children five,
In the nest and all alive,
Mr. Spikky Sparrow stood;
Mrs. Sparrow sate close by,
On a little piece of wood,
Singing with a cheerful smile
Spikky bikky bee!
To amuse them all the while,
Twikky wikky wikky wee,
Wikky bikky twikky tee,
```

Hint: use the sentinel to save all the input strings into a list, then sort it with a method.

3.3 Problem 2 (1 mark)

This problem and the rest are due at the beginning of the next workshop. *Ask any questions you have about the problems NOW!*

Problem: The *median* of a sequence of numbers is the value in the middle, that is, if the numbers are sorted, as many numbers in the sequence are above the median as below. If there are an odd number of values, the median is just the middle one. If there is an even number of values, the median is half way between the two values closest to the middle.

Write a program that allows the user to enter numbers until a blank line is entered instead of number, and then prints the median. For example:

```
Enter a number: 2
Enter a number: 3
Enter a number: 1
Enter a number:
Median = 2.0
```

```
Enter a number: 2
Enter a number: 4
Enter a number: 1
Enter a number: 3
Enter a number:
Median = 2.5
```

3.4 Problem 3 (2 marks)

Problem: A *palindrome* is a string that reads the same backwards as forwards. It is usual to remove all non-letters from the string first and to ignore case.

Write a *function* that accepts a string as its argument and returns `True` if and only if the string is a palindrome.

Your main program should allow the user to test input strings until they enter an empty one. For example:

```
Enter a string: abc
It is not a palindrome.
Enter a string: Abba
It is a palindrome!
Enter a string: canoe
```

```
It is not a palindrome.
Enter a string: Kayak
It is a palindrome!
Enter a string: Madam, I'm Adam.
It is a palindrome!
Enter a string: Was it a car or a cat I saw?
It is a palindrome!
Enter a string:
```

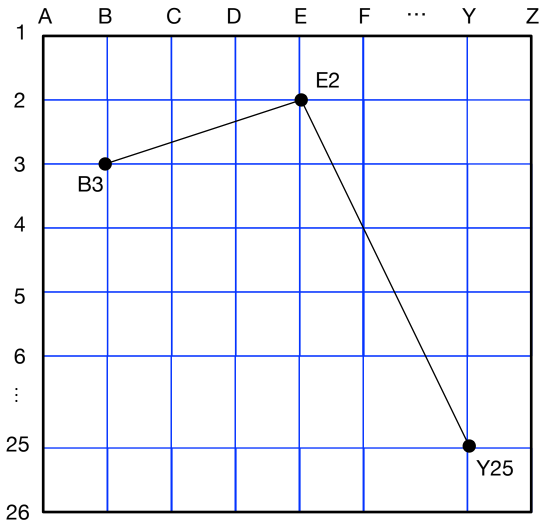
Hints: The function might use a for loop to build up a list of only the letters in upper case. A second for loop can check that the first letter equals the last, the second equals the second last, etc. Python makes this easy with its negative indices.

For your mild amusement.

3.5 Problem 4 (1 marks)

This is meant to be a challenging problem that only the best students will succeed with at this stage of the course.

Problem: A road map defines locations as map references like B3, where B is the x-coordinate value and 3 is the y-coordinate.



The grid lines are 0.5 km apart.

Write a program that allows the user to enter a trip as a sequence of any number of map references on one line, and reports the total length of the trip, assuming they can travel in straight lines. For example:

```
Enter trip map references: C2 B5 Y25
Total distance = 16.8 km
```

For badly formatted map references, your program should exit, reporting the first bad map reference.

```
Enter trip map references: E6 E4 D7 d43 F5
Bad reference:  d43
```

Hints: you need to *split* the input line into separate references; each reference starts with one character which must be an upper case letter, and the rest must be only digits; and Pythagoras will help. The function `exit()` can abort the program if you detect an error in the input.

4 After the workshop

- You have created programs that might be useful to refer back to in future workshops. Make sure that you will have that work in the future. One copy is not enough for at IT professional. You should have at least 2 copies:

  1. on your Griffith network storage drive; and
  2. on your portable storage device.

- Please answer these feedback questions.

  - What was the best aspect of this workshop?
  - What was the most difficult or worst aspect of this workshop?
  - Did you find an error in these workshop notes?