

CS 115 - Introduction to Programming in Python

Lab 05

Lab Objectives: Tuples, Lists, Dictionaries

Notes:

1. Upload your solutions as **a single .zip file** to the Lab05 assignment for your section on Moodle. You must use the following naming convention: Lab05_Surname_FirstName.zip where Surname is your family name and FirstName is your first name.
2. You should only use functionality covered in CS115 in your solution.
3. Include a docstring for your functions.

1. a) Write a function `countPeaks()` that takes a tuple as parameter, finds and returns the number of peak values in the tuple. A peak value is greater than its predecessor and its successor.

Assume that the tuple contains numbers.

Example: (4, 2, 5, 7, 6, 6, 4, 3, 5, 2, 6, 7) has 2 peaks, the first 7 and the second 5. Both of these values are greater than their predecessor and their successor.

b) Write a script (**Lab5Q1.py**) to input the number of elements `n` in a tuple and then input `n` integer elements of the tuple and display the number of peak values in the tuple by the `countPeaks()` function.

Sample Run:

```
Enter the number of elements in the tuple: 8
Enter element 1 : 23
Enter element 2 : 15
Enter element 3 : 18
Enter element 4 : 9
Enter element 5 : 3
Enter element 6 : 5
Enter element 7 : 7
Enter element 8 : 4
There are 2 peak values in (23, 15, 18, 9, 3, 5, 7, 4)
```

2. a) Write a function **shortest** which takes a list of strings as a parameter, finds and returns the shortest string in the list. The shortest word will be deleted from the list.

If there are more than one words that have same smallest length, than the function will return the first

- b) Write a script (**Lab5Q2.py**) that initializes a list of strings, displays the original list and then displays the shortest word in the list and the updated list after calling the **shortest** function.

Sample Run:

Original List:

```
['abc', 'qwer', 'ss', 'x123', 'nn', 'at', '4321']
```

Shortest word: ss

New List:

```
['abc', 'qwer', 'x123', 'nn', 'at', '4321']
```

3. a) Write a function **countLetters** which takes a string and returns the dictionary where the keys are each unique letters in the string and the values are the number of occurrences of that letter in the string.

b) Write a script (**Lab5Q3.py**) to input a string and then call the above **countLetters** function to display the contents of the dictionary produced by the function into a file **out.txt**

Sample Run:

Enter a string: **this is a dog**

out.txt will contain:

t	1
h	1
i	2
s	2
	3
a	1
d	1
o	1
g	1