CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

COMP7044 - Systems Scripting

Semester 2, 2020

Assignment 2: Python Scripting

Submission Deadline: Thursday 30th April 2020 Midnight

For each task write a short report describing your solution concept and how you arrived at it. This should be done as a block comment on top of each task solution script.

Task 1

Write three Python functions that accept a list of string items as parameter. The first function should return the first item in the list whose length is divisible by 3. The second function should return the last item in the list with more than 4 characters. The third function should replicate the list by 3 and output one entry per line all the items with odd number of characters.

In the main program, interactively create a list of 5 string items (words, sentences etc) of varying length. Present the user with a set of menu to select from. If the user enters the character 'F' or 'f', the program should output the first item whose length is divisible by 3. When the user enters the character 'L' or 'l', the program should output the last item with more than 4 characters. When the user enters any number, the program should perform the replication and output the items with odd number of characters. The program should go back to the menu options for new selection. When the user enters 'E' or 'e', the program should terminate with a goodbye message.

[10 marks]

Task 2

Write a Python function that accepts two parameters. The first parameter should be a list of items and the second parameter should be an integer value representing the number of indexes. Use a random generator to generate indexes in the range of 2 to 5 based on the second parameter. Use the generated indexes to randomly remove from the list the items that corresponds to those indexes. The function should return the remaining items as a tuple.

Interactively create a list of words or numbers that has at least 12 items. Then request a user to enter the number of items to be deleted. This number should fall between 2 and 6. Invoke your function with the appropriate parameters. Output per line, first the returned tuple and then the original lists.

Sample program output when 4 items were randomly deleted:
Result tuple: (4, 5, 3, 8, 9, 5, 3, 5, 9)
Original List: [4, 5, 6, 3, 2, 8, 9, 3, 5, 4, 3, 5, 9]

[8 marks]

Task 3

Write a function named *reducer()* that accepts an integer parameter named *number*. If *number* is even then *reducer()* should divide *number* by 2 and return this value. If *number* is odd, then reducer should return 3 times *number* + 1.

Then write a program that requires a user to enter an integer number and that keeps calling *reducer()* on that number until the function returns the value 1. (Amazingly, this sequence works for any integer value. Sooner or later you will arrive at value 1). Example output sequence for entering the number 3 is:

```
10
5
16
8
4
2
1
```

[7 marks]

Task 4

Write a Python script that operates in two phases. The first phase should automate the creating of a folder called “task4”. If this folder exist, delete it and recrate it. Inside this folder, create two subfolders named “backup” and “working”. Inside the “working” folder create three other subfolders named “pics”, “docs” and “movie”. Inside the “docs” folder create five files (CORONAVIRUS.txt, DANGEROUS.txt, KEEPSAFE.txt, STAYHOME.txt, HYGIENE.txt) with varying content of your choice and two subfolders (school and party).

In the second phase, the script should rename all the files in the “docs” folder to lowercase. The extension “.txt” should be renamed to uppercase. Note that the subfolders in that directory should remain unchanged. When the renaming is complete, the program should use the Python zipfile module to archive the “docs” folder and make five backup archives of it in the top-level “backup” folder. Output the content of the backup folder and one of the zip archives for verification purpose.

[10 marks]

Submission Instruction:

Each task should be solved with a different script file. Put the solution scripts into a folder including the folder structure for Task4 and name the folder as follows: **firstname-surname**. Create a Zip archive from the folder. It should have the name: **firstname-surname.zip** E.g., Vincent-Emeakaroha.zip. Upload this archive to Canvas.