MTU
Ollscoil Teicneolaíochta na Mumhan
Munster Technological University

COMP7044 - Systems Scripting

Semester 2, 2021

Assignment 2: Python Scripting

Submission Deadline: Friday 23th April 2021 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 a Python script that implements two functions where each accepts a single list parameter containing strings. The first function should check and output the list items containing a question mark. The second function should check and output all characters that appear in each of the list items.

Interactively create a list with the following content to test your functions:

"farshad", "ghassemi?d", "madam", "?radar?", "duration", "con?tained"

Sample output:

```
Question marks check:
ghassemi?d contains question mark
?radar? contains question mark
con?tained contains question mark

Common character check:
Character  a  appears in all items
farshad contains  2  a
ghassemi?d contains  1  a
madam contains  2  a
?radar? contains  2  a
duration contains  1  a
con?tained contains  1  a
Character  d  appears in all items
farshad contains  1  d
ghassemi?d contains  1  d
madam contains  1  d
?radar? contains  1  d
duration contains  1  d
con?tained contains  1  d
```

[28 marks]

Task 2

Write a Python script that implements a function that accepts two parameters. The first parameter should be a list of any items and the second parameter should be an integer value. Use a random generator to generate a number of indexes corresponding to the value of the second parameter within the range of 1 to 4. Remove items from the given list (first parameter) based on the randomly generated indexes. The function should return the final list 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. Ensure that this ranged is enforced and handle exceptions. 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', 'her', '3', '9', 'he', '7', '6', '2')
Original List: ['4', '5', 'me', 'her', '3', '8', 'see', '9', 'he', '7', '6', '2']

[22 marks]

Task 3

Write a Python script that implements a recursive 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 request a user to enter an integer number and recursively call *reducer()* on that number until the function returns the value 1. Use exception handling to ensure that user enters an integer number before proceeding. (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

[20 marks]

Task 4

Write a Python script that implements functions to address the following task. The first function should accept a string parameter representing a folder name. This folder name should be interactively provided by the user.

The first function should automate the creation of a folder structure starting with the provided folder name. 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).

Use another function to rename all the files in the “docs” folder to lowercase. The extension “.txt” should be renamed to uppercase. Ensure that the folder exist before proceeding and note that the subfolders in that directory should remain unchanged.

When the renaming is complete, use another function to implement 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.

[30 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 Task 4 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.