Design and Analysis of Software Systems – Sprint 2024
Assignment 2 – Software Testing
Due Date: 4 March 2024, 8.00 PM

Instructions:

- This assignment is an individual submission.
- Total Marks of 50 Marks for 3 weeks.
- All script submissions should be submitted via Moodle.
- Inputs/output should fit the criteria mentioned in respective question.
- All other conditions are open to your interpretations.
- Please ensure that you maintain a clear record of your submission by following the submission criteria with clear code changes and clean organization of files.
- For queries, reach out to TAs via Moodle.

Q1: Using Pylint module in Python, conduct static code analysis or lint following python programs. Share the results as per the recommendation below:

a. From Assignment 1, pick your **kaooa board game** and lint the program. Submit the following: **(10 Marks)**
   - Submit the initial lint result along with Pylint score.
   - Review the recommendations from your lint results, address them in your code and capture the lint results.
   - Repeat linting by addressing as many recommendations as possible and capture the lint results along with score after every iteration.

b. Consider L_n as integer part of a series L. when **n=0** then L_n **= -3**, and L_n **= -1** when **n=1** and $L_n = L_{n-1} + L_{n-2}$ when **n > 1.** Now write a Python Script that takes input **n** and returns its corresponding number L_n value as output. Assume the input is **1 < n < 1000001.** Example: if **n = 2** then L_2 **= -4** . Later using Pylint, address as many recommendations as possible and submit your best pylint score, its related report and code version. **(10 Marks)**

Q2: Using **unittest** – a Unit Testing Framework in Python, build unit test cases for following routine.

a. Pick any four-digit number as input, using at least two different digits (leading zeros are allowed). Exclude 1111 or 0000 as your input.
b. Arrange the digits in descending and then in ascending order to get two four-digit numbers, adding leading zeros if necessary.
c. Subtract the smaller number from the bigger number
d. Go back to step 2 and repeat until you reach a constant i.e. 6174

Example: **Input** = 9218 **Output:** 9218, 8532, 6174. **Computation**: Rearrange 9821, 1289, then 9821-1289=8532. Repeats rearrangement 8532 as 8532, 2358, then 8532-2358=6174.
Write a Python program to process the above routine. Now write at least 10 test cases of your choice using different assert methods and document your results. **(10 Marks)**

Q3: Write a Python Script to print Palindrome days in DD-MM-YYYY when the year number is provided as input.

Example: if Input Year=2001, Output = 10-02-2001. If there are no palindrome days in DD-MM-YYYY, print **"No Palindrome days available in the given year"**

Later using PyTest, write at least 5 test cases of your choice using assert. **(10 Marks)**

Q4: You are tasked with creating a program that analyses student scores. Each student is represented as a dictionary with the following key - 'name' and value - 'scores. The 'scores' is a list of integers representing the students' scores in different subjects. Write a python program that does the following:

1. Define a function called calculate_average that takes a list of student dictionaries as input and returns a new dictionary where the keys are the student names, and the values are the average scores of each student/
2. Implement a function called find_highest_scorer that takes list of student dictionaries as input and returns following:
    a. The name of the student with highest average score.
    b. Ordered list of student names who scored the highest in each corresponding subject (like firstname corresponds to the student scoring the highest in the first subject).
3. In the main part of the program, take the user input to enter information for several students. Use the functions defined above to calculate and display the average scores of each student and identify the student with the highest average score & highest score in individual subjects. (Menu Based)

You may have to create your own dataset, do not share or re-use dataset created by others and do appropriate error handling. Now use **unittest** or **PyTest** to write at least 3 testcases for each requirement i.e. 4.1, 4.2 and 4.3. **(10 Marks)**

Resources:

- Pylint - https://pylint.pycqa.org/en/v3.0.3/index.html
- unitest - https://docs.python.org/3/library/unittest.html
- PyTest - https://pytest.org/en/latest/how-to/assert.html

Submission Criteria:

Q1.a - Submit in following format in master folder q1a

- Folder: **OriginalGame** -> place your kaooa game python script
- Folder: **InitialLint** -> place your initial lint result in text file

- Folder: **AllLint** -> include all kaooa game revisions along with corresponding lint results. Name the final version as kaooafinal.py and lintfinal.txt

Q1.b - Submit in following format in master folder q1b

- Folder: **Code** -> contains initial version of your script as Lucasinitial.py. Name the rest of the revised versions as Lucas_1.py, Lucas_2.py etc.
- Folder: **Lint** -> contains initial lint file as lintinitial.txt. Name the rest of the lint files as lint_1.txt, lint_2.txt to corresponding revised code in **Code** folder

Q2 – Submit the following format in master folder q2

- Folder: **Code** -> contains your kaprekarroutine.py
- Folder: **testcases** -> contains all your testcases associated with your code.

Q3- Submit the following format in master folder q3

- Folder: **Code** -> contains your palindrome.py
- Folder: **testcases** -> contains all your testcases associated with your code.

Q4- Submit the following format in the master folder q4

- Folder: **Code** -> one python program to include 4.1, 4.2 and 4.3
- Folder: **Dataset** -> randomly generated dataset. Include a file for dataset generation.
- Folder: **testcases**-> contains all your testcases associated with 4.1, 4.2, 4.3

README.md* - This is a mandatory file, please write notes about all questions (what test cases you have written, why, etc.) and instructions on how to run their unit tests/programs etc. Any other notes to evaluators can be described here.

Add all the folders q1a, q1b, q2, q3, q4 and README.md file in a directory named <rollnumber> and zip the contents as <rollnumber>.zip. If and only if the size of your submission is more than 20 MB, please check-in your assignment submission in github and provide us the repository URL in readme.txt. ZIP the readme.txt in <rollnumber>.zip

P.S.: Use of Code Assist tools and GENERATIVE AI is not permitted.