CSC 265: Assignment 3

Dates

- **Steps A and B (unmarked)**: in lab the week of 16 November
- **Steps A, B, C, and D (marked)**: in lab the week of 23 November
- **Final submit deadline**: Saturday 28 November at 10:00 AM

Summary

In each of the weekly lab quizzes, the quiz results are stored in XML log files. In each quiz, there is a separate log file for each student.

Your task is to develop three Python applications as well as a shared library:

- `quiz_marks.py` takes a list of log files and computes the quiz mark for each log file.
- `quiz_pass_ratio.py` takes a list of log files and computes the pass ratio for each quiz question.
- `quiz_time.py` takes a list of log files and computes the average time taken for each quiz question.
- `quiz_library.py` provides functions used by the three applications.

Download and unpack the assignment files

1. **Download**. Use the web browser to download assignment_03.zip and save it in your home directory.

2. **Unpack**. Right click on the zip file and select extract.

Quiz log files

A *quiz log file* is an XML file containing a record of the answers entered and buttons pressed by one student during a quiz.

In a *legal quiz log file*, the root tag must be named `quiz_log`. Following the root tag, there must be XML elements of two kinds: `answer` and `display`.

In each `answer` element there must be:

- an `index` element, whose content must be a non-negative integer,
- a `path` element,
- a `result` element, whose content must be either a non-negative integer, or empty,
- an `answer` element,
- a `time` element, whose content must be either a non-negative integer, or empty.

The `index`, `path`, `result`, `answer` and `time` elements may occur in any order within the `answer` element.

In each `display` element there must be:

- an `index` element, whose content must be a non-negative integer,
- a `path` element,
- a `time` element, whose content must be a non-negative integer.

The `index`, `answer` and `time` elements may occur in any order within the `display` element.

A legal quiz log file for a quiz with *N* questions begins with *N* `answer` elements. The first `answer` element has `index` value 0, the second `answer` element has `index` value 1, and so on. In these `answer` elements, the `result`, `answer`, and `time` elements all have empty content.

Following the N answer elements there must be a display element, with index content 0 and time content equal to the time the student started the quiz.

The rest of the log file contains a mixture of zero or more `answer` and `display` elements, with `index` content in [0..N-1] and non-empty `result` content.

Deliverable: `quiz_library.py`

Specification

The correct behaviour of the functions is specified in `quiz_library.py`.

Implementation

Your code must be correct with respect to the specifications in `quiz_library.py`.

Your implementation must be contained **completely** in the bodies of the functions contained in `quiz_library.py`. **Do not modify** any other part of `quiz_library.py`.

Deliverable: `quiz_marks.py`

Specification

The correct behaviour of this application is specified in `quiz_marks.py`.

Implementation

Your code must be correct with respect to the specification in `quiz_marks.py`.

Deliverable: `quiz_pass_ratio.py`

Specification

The correct behaviour of this application is specified in `quiz_pass_ratio.py`.

Implementation

Your code must be correct with respect to the specification in `quiz_pass_ratio.py`.

Deliverable: `quiz_time.py`

Specification

The correct behaviour of this application is specified in `quiz_time.py`.

Implementation

Your code must be correct with respect to the specification in `quiz_time.py`.

Development steps

Step A

- Implement the three functions in `quiz_library.py`.
- Test the code:

```
python test_quiz_library.py x0.xml > test_quiz_library_x0.txt
diff test_quiz_library_x0.txt test_quiz_library_gold_x0.txt

python test_quiz_library.py x1.xml > test_quiz_library_x1.txt
diff test_quiz_library_x1.txt test_quiz_library_gold_x1.txt
```

- There should be no differences reported.

Step B

- Implement `quiz_marks.py`.
- Test the code:

```
python quiz_marks.py x1.xml x2.xml > test_quiz_marks.txt
diff test_quiz_marks.txt test_quiz_marks_gold.txt
```

- There should be no differences reported.

Step C

- Implement `quiz_pass_ratio.py`.
- Test the code:

```
python quiz_pass_ratio.py x1.xml x2.xml > test_quiz_pass_ratio.txt
diff test_quiz_pass_ratio.txt test_quiz_pass_ratio_gold.txt
```

- There should be no differences reported.

Step D

- Implement `quiz_time.py`.
- Test the code:

```
python quiz_time.py x1.xml x2.xml > test_quiz_time.txt
diff test_quiz_time.txt test_quiz_time_gold.txt
```

- There should be no differences reported.