CSE-314 Offline-1
Assignment On Shell Scripting
Deadline: 11:59 pm, 05 Dec 2022

Your job is to write a bash script that evaluates each student's script, compares the output with an expected output text**,** and grades the script accordingly. You will also write a "copy-checker" (of course, a silly one!) to detect any plagiarism.

Input:

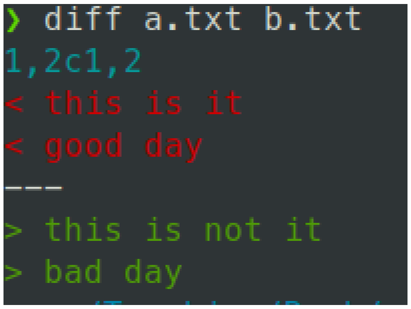
You will be given a zip folder. After unzipping it, the contents of the directory will contain:

1. **Submissions** directory will contain the submissions of all the students. Under the **Submissions** directory, there will be a directory for each student who submitted the assignment. Each student directory must contain a **sh** file named **<student_id.sh>**. This is the script that will be executed for evaluation. For example, **Submissions/1805123/1805123.sh** is the submission script of student ID 1805123.
2. **AcceptedOutput.txt** is a text file that contains the expected output from students' solutions.

Task Details:

1. Your script will take two arguments:
    a. **max_score:** a **positive integer** argument that will be the **maximum score** a student can get. If no argument is provided, use **100** as the default value
    b. **max_student_id**: a **positive integer** argument between **1 to 9 (inclusive)** that determines the last considered student's ID. For example, if the **max_student_id** is 5, the student IDs will be 1805121, 1805122, 1805123, 1805124, and 1805125. If the **max_student_id** is 9, the student IDs will be 1805121, 1805122, ……, 1805129. Use **5** as the **default value**.
2. Run the student's **<student_id>.sh** file, using the student's folder as the working directory. (You may want to capture the student's output into a file.) Note that the student script files may not have execute permission, so you should manually invoke bash to run them, such as **bash ./script.sh**

3. Compare the student’s output against the expected output file using **diff**. The expected output is **case-sensitive**. Run diff so that it ignores all white space (see the **man** page of **diff**). Count the number of lines that do not match. We will keep the definition of mismatch simple: we will consider any line of **diff** output containing a **<** or a **>** to count as 1 line of unmatched content.

   ```
   ❯ diff a.txt b.txt
   1,2c1,2
   < this is it
   < good day
   ---
   > this is not it
   > bad day
   ```

   The image above has a mismatch of 4.

4. For each unmatched line, deduct **5** points from the student’s score. For example, if the diff output has 4 lines that have < or > (the above case), the student should lose 20 scores. If the student loses more points than there are points in the assignment, the student should receive **0** points.
5. If there is no directory for a **student_id** or the directory does not contain the <student_id>.sh file or incorrectly named the file, the student should get **0** points on the assignment.
6. Now, if a student is caught in the copy-checker, his/her score will be **negative** of his/her **initial score**. If the initial score was 60, now it will be -60. The definition of copy is somewhat loose here. We will call two scripts copies of each other if they exactly match i.e: the **diff** command produces no output. You should ignore **trailing whitespaces** and **blank lines** while comparing.
7. You can create any temporary files as needed, but they should not persist after your script ends.
8. You can assume the students’ scripts are not malicious and they don’t hang indefinitely.

Output:

You have to prepare a CSV file named **output.csv** with two columns **student_id**, and **score**.

Help:

- Have a look at the **man** page of **diff.**
- The following link might help work with CSV files: https://bconnelly.net/working-with-csvs-on-the-command-line/

Restrictions:

Please DO NOT COPY solutions from anywhere (your friends, seniors, internet, etc.). Any form of plagiarism (irrespective of source or destination), will result in getting -100% marks in this assignment. You have to protect your code.

Disclaimer:

The sample input and output files are simply for clarifying the formats with parameters **max_score** = 50 and **max_student_id** = 5. No guarantee is given that the outputs will be the same for the provided input.

Submission Guideline:

1. Create a directory with your 7-digit student id as your name
2. Rename your shell script file with your 7-digit student id
3. Put this script into the directory created in 1
4. Zip the directory
5. Upload the zip into moodle

For example, if your student id is 1805123, create a directory named 1805123, and rename your .sh file into 1805123.sh. Put 1805123.sh into 1805123, and then zip 1805123 into 1805123.zip, and upload the 1805123.zip into moodle.

Failure to follow the above-mentioned submission guideline will result in some penalties.