Introduction

This is a simple script that helps to test your code input/output format. A similar approach will be used to test your submission. If your code runs with this script it is more likely to run without any changes when we test.

Note: This script only checks the input/output format of your code and compilation.

This script supports only mac and Linux. To test this in windows, you can use git bash. Please follow the below link for git bash https://stackoverflow.com/a/37478310/3925240

Script Setup

1. Copy the script **test_aoa_pa2.sh** into some folder. Here we have copied to the folder **test**.

```
revanth@Revanths-MacBook-Air test % ls -l
total 16
-rw-r--r--@ 1 revanth  staff  5123 Apr 29 16:38 test_aoa_pa2.sh
revanth@Revanths-MacBook-Air test %
```

2. Inside this folder make a directory named PA2-data-public.

```
revanth@Revanths-MacBook-Air test % ls -l
total 16
-rw-r--r--@ 1 revanth  staff  5123 Apr 29 16:38 test_aoa_pa2.sh
revanth@Revanths-MacBook-Air test %
revanth@Revanths-MacBook-Air test %
revanth@Revanths-MacBook-Air test % mkdir PA2-data-public
revanth@Revanths-MacBook-Air test %
revanth@Revanths-MacBook-Air test % ls -l
total 16
drwxr-xr-x  2 revanth  staff    64 Apr 29 16:41 PA2-data-public
-rw-r--r--@ 1 revanth  staff  5123 Apr 29 16:38 test_aoa_pa2.sh
revanth@Revanths-MacBook-Air test %
```

3. Extract the contents of test cases shared in piazza
https://piazza.com/class_profile/get_resource/kyzxm91gia64r8/l1vcagm69lg6g5

4. After extracting, the contents of **PA2-data-public** have to be something like this.

```
revanth@Revanths-MacBook-Air PA2-data-public % ls -l
total 96
-rw-r--r--@  1 revanth  staff  5203 Apr 29 16:14 PA2-Testcases-Public.zip
drwxr-xr-x@ 12 revanth  staff   384 Apr 29 16:15 __MACOSX
-rw-r--r--@  1 revanth  staff    12 Nov 27  2016 input1.txt
-rw-r--r--@  1 revanth  staff    24 Nov 27  2016 input2.txt
-rw-r--r--@  1 revanth  staff    27 Nov 27  2016 input3.txt
-rw-r--r--@  1 revanth  staff   302 Nov 27  2016 input4.txt
-rw-r--r--@  1 revanth  staff   352 Nov 27  2016 input5.txt
-rw-r--r--@  1 revanth  staff     6 Nov 27  2016 output1.txt
-rw-r--r--@  1 revanth  staff    12 Nov 27  2016 output2.txt
-rw-r--r--@  1 revanth  staff    10 Nov 27  2016 output3.txt
-rw-r--r--@  1 revanth  staff    96 Nov 27  2016 output4.txt
-rw-r--r--@  1 revanth  staff   156 Nov 27  2016 output5.txt
```

5. Copy your code file and LCS_checker.py into the folder **test** where you have copied the test script.
LCS_checker.py
https://piazza.com/class_profile/get_resource/kyzxm91gia64r8/l259o82im7n3s4

```
revanth@Revanths-MacBook-Air test % ls -l
total 32
-rw-r--r--@ 1 revanth  staff   911 Apr 29 16:42 LCS_checker.py
-rw-r--r--  1 revanth  staff  1829 Apr 29 16:42 LongestCommonSubsequence.java
drwxr-xr-x  2 revanth  staff    64 Apr 29 16:41 PA2-data-public
-rw-r--r--@ 1 revanth  staff  5123 Apr 29 16:38 test_aoa_pa2.sh
revanth@Revanths-MacBook-Air test %
revanth@Revanths-MacBook-Air test %
```

6. Now execute the command **bash test_aoa_pa2.sh <Program type> <Filename>**

Example: **bash test_aoa_pa2.sh JAVA LongestCommonSubsequence.java**

```
-- Usage: bash test_aoa_pa2.sh <Program type> <Filename>

   Program type - JAVA/PYTHON/CPP

   Filename - Name of the file name that has to be tested.

-- Example:

   bash test_aoa_pa2.sh JAVA PA1.java
```

Assumptions

1. We assume the submission is a single file. If you have multiple files, merge them into one.

2. Also, you don't need any external libraries for the assignment. So the commands we use for compilation are simple and straightforward.

   JAVA:
   Compile: javac <filename>
   Execution: java <filename_without_extension>

   C++
   Compile: g++ -std=c++11 -o <executable_name> <filename>
   Execution: ./<executable_name>

   For python we use python3 to run your script. **python3 <filename>**

   The above commands should cover most cases. If you need to add some flags for compilation or use a different command for execution, raise a request in Piazza. The same will be added to the script if it is a valid request.

Validation

If you get below output, your input/output processing is correct.

```
revanth@Revanths-MacBook-Air test % bash test_aoa_pa2.sh JAVA LongestCommonSubsequence.ja
rm: LongestCommonSubsequence.class: No such file or directory

-------Compiling the code-------
Success

-------Testing public test cases-------

1.) Testing File: PA2-data-public/input1.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input1.txt > temp/output1.txt
Execution time:
        1 s.
---------------------
2.) Testing File: PA2-data-public/input2.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input2.txt > temp/output2.txt
Execution time:
        0 s.
---------------------
3.) Testing File: PA2-data-public/input3.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input3.txt > temp/output3.txt
Execution time:
        0 s.
---------------------
4.) Testing File: PA2-data-public/input4.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input4.txt > temp/output4.txt
Execution time:
        0 s.
---------------------
5.) Testing File: PA2-data-public/input5.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input5.txt > temp/output5.txt
Execution time:
        0 s.
---------------------

******* All tests passed. *******

revanth@Revanths-MacBook-Air test %
```

If you get something like the below output, there are 2 cases of failure

1. There are some extra output getting printed
2. The actual output does not meet the expected. The same can be checked with the diff command printed in the console or LCS_checker.py

```
revanth@Revanths-MacBook-Air test %
revanth@Revanths-MacBook-Air test % bash test_aoa_pa2.sh JAVA LongestCommonSubsequence.java

-------Compiling the code-------
Success

-------Testing public test cases-------

1.) Testing File: PA2-data-public/input1.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input1.txt > temp/output1.txt
Execution time:
        0 s.
[FAILED] Incorrect subsequence length for PA2-data-public/input1.txt.
         Use 'diff <(head -n 1 PA2-data-public/output1.txt) <(head -n 1 temp/output1.txt)' to know the diff

[FAILED] Output subsequence is incorrect for PA2-data-public/input1.txt.
         Use 'python2 LCS_checker.py PA2-data-public/input1.txt PA2-data-public/output1.txt temp/output1.txt' to know the failure reason

---------------------
2.) Testing File: PA2-data-public/input2.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input2.txt > temp/output2.txt
Execution time:
        0 s.
[FAILED] Incorrect subsequence length for PA2-data-public/input2.txt.
         Use 'diff <(head -n 1 PA2-data-public/output2.txt) <(head -n 1 temp/output2.txt)' to know the diff

[FAILED] Output subsequence is incorrect for PA2-data-public/input2.txt.
         Use 'python2 LCS_checker.py PA2-data-public/input2.txt PA2-data-public/output2.txt temp/output2.txt' to know the failure reason

---------------------
3.) Testing File: PA2-data-public/input3.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input3.txt > temp/output3.txt
Execution time:
        0 s.
[FAILED] Incorrect subsequence length for PA2-data-public/input3.txt.
         Use 'diff <(head -n 1 PA2-data-public/output3.txt) <(head -n 1 temp/output3.txt)' to know the diff

[FAILED] Output subsequence is incorrect for PA2-data-public/input3.txt.
         Use 'python2 LCS_checker.py PA2-data-public/input3.txt PA2-data-public/output3.txt temp/output3.txt' to know the failure reason

---------------------
4.) Testing File: PA2-data-public/input4.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input4.txt > temp/output4.txt
Execution time:
        0 s.
[FAILED] Incorrect subsequence length for PA2-data-public/input4.txt.
         Use 'diff <(head -n 1 PA2-data-public/output4.txt) <(head -n 1 temp/output4.txt)' to know the diff

[FAILED] Output subsequence is incorrect for PA2-data-public/input4.txt.
         Use 'python2 LCS_checker.py PA2-data-public/input4.txt PA2-data-public/output4.txt temp/output4.txt' to know the failure reason

---------------------
5.) Testing File: PA2-data-public/input5.txt
Executing:
        java LongestCommonSubsequence < PA2-data-public/input5.txt > temp/output5.txt
Execution time:
```

The script should run in 1 - 2 secs. If the script hangs, then your input/output processing is wrong.