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.

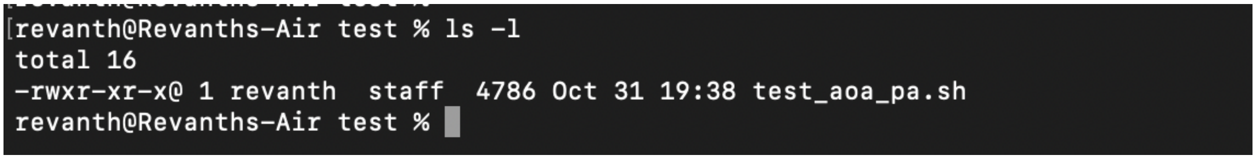
Note: This script only checks your code's input/output format and compilation.

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

If you face any issues, create a private Piazza post to Revanth.

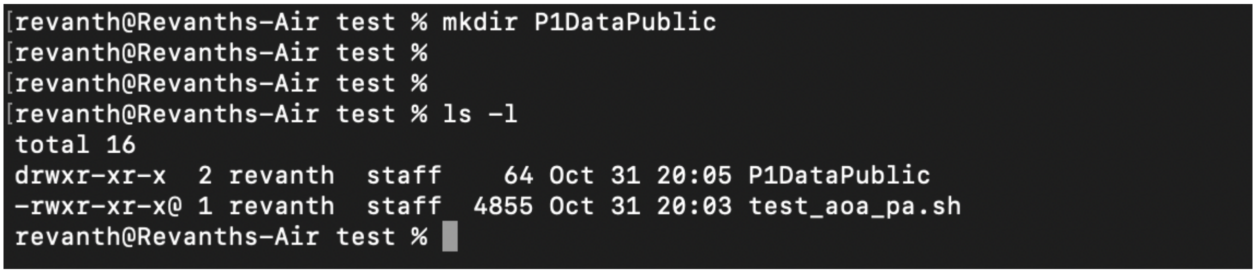
Script Setup

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



```
[revanth@Revanths-Air test % ls -l
total 16
-rwxr-xr-x@ 1 revanth  staff  4786 Oct 31 19:38 test_aoa_pa.sh
revanth@Revanths-Air test % 
```

2. Inside this folder make a directory named **P1DataPublic**

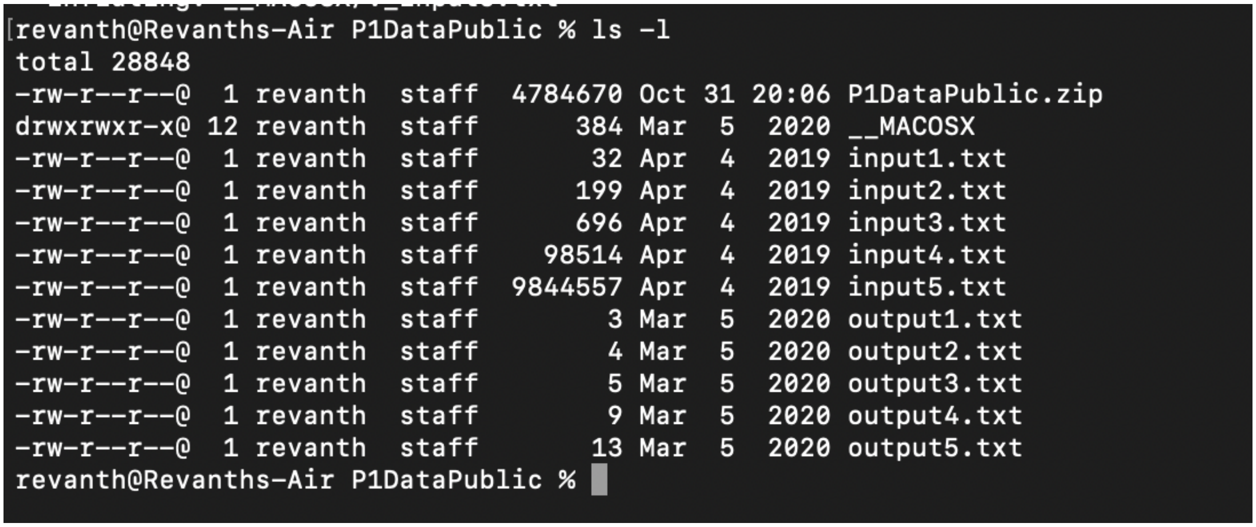


```
[revanth@Revanths-Air test % mkdir P1DataPublic
[revanth@Revanths-Air test %
[revanth@Revanths-Air test %
[revanth@Revanths-Air test % ls -l
total 16
drwxr-xr-x  2 revanth  staff    64 Oct 31 20:05 P1DataPublic
-rwxr-xr-x@ 1 revanth  staff  4855 Oct 31 20:03 test_aoa_pa.sh
revanth@Revanths-Air test % 
```

3. Now extract the contents of test cases shared in piazza

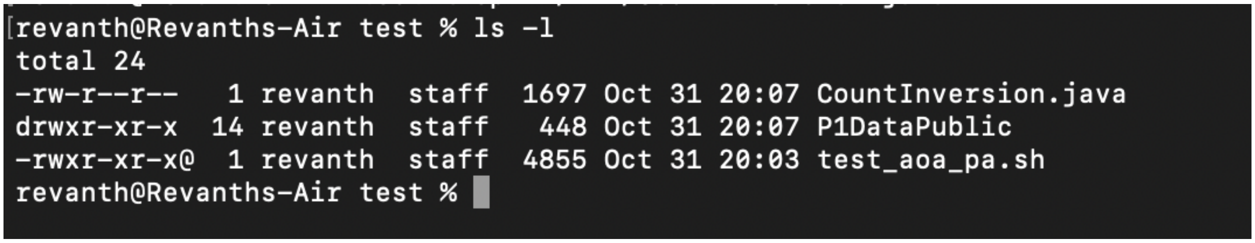
https://piazza.com/class_profile/get_resource/l7aktlpowdz24l/l98i3w3wecn110

4. After extracting, the contents of **P1DataPublic** have to be something like this.



```
[revanth@Revanths-Air P1DataPublic % ls -l
total 28848
-rw-r--r--@  1 revanth  staff  4784670 Oct 31 20:06 P1DataPublic.zip
drwxrwxr-x@ 12 revanth  staff      384 Mar  5  2020 __MACOSX
-rw-r--r--@  1 revanth  staff       32 Apr  4  2019 input1.txt
-rw-r--r--@  1 revanth  staff      199 Apr  4  2019 input2.txt
-rw-r--r--@  1 revanth  staff      696 Apr  4  2019 input3.txt
-rw-r--r--@  1 revanth  staff    98514 Apr  4  2019 input4.txt
-rw-r--r--@  1 revanth  staff  9844557 Apr  4  2019 input5.txt
-rw-r--r--@  1 revanth  staff        3 Mar  5  2020 output1.txt
-rw-r--r--@  1 revanth  staff        4 Mar  5  2020 output2.txt
-rw-r--r--@  1 revanth  staff        5 Mar  5  2020 output3.txt
-rw-r--r--@  1 revanth  staff        9 Mar  5  2020 output4.txt
-rw-r--r--@  1 revanth  staff       13 Mar  5  2020 output5.txt
revanth@Revanths-Air P1DataPublic %
```

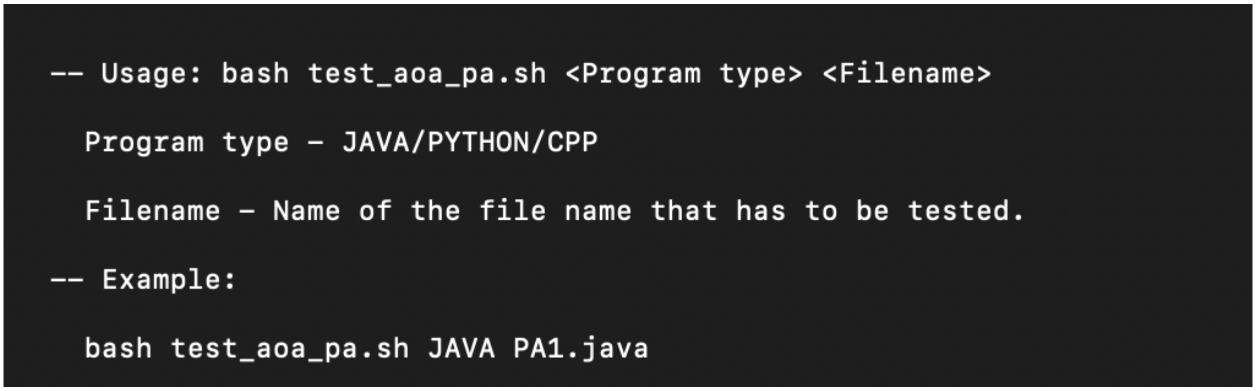
5. Now copy your code file into the folder where you have copied the test script.



```
[revanth@Revanths-Air test % ls -l
total 24
-rw-r--r--   1 revanth  staff  1697 Oct 31 20:07 CountInversion.java
drwxr-xr-x  14 revanth  staff   448 Oct 31 20:07 P1DataPublic
-rwxr-xr-x@  1 revanth  staff  4855 Oct 31 20:03 test_aoa_pa.sh
revanth@Revanths-Air test %
```

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

Example: **bash test_aoa_pa.sh JAVA PA1.java**



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

  Program type - JAVA/PYTHON/CPP

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

-- Example:

  bash test_aoa_pa.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>

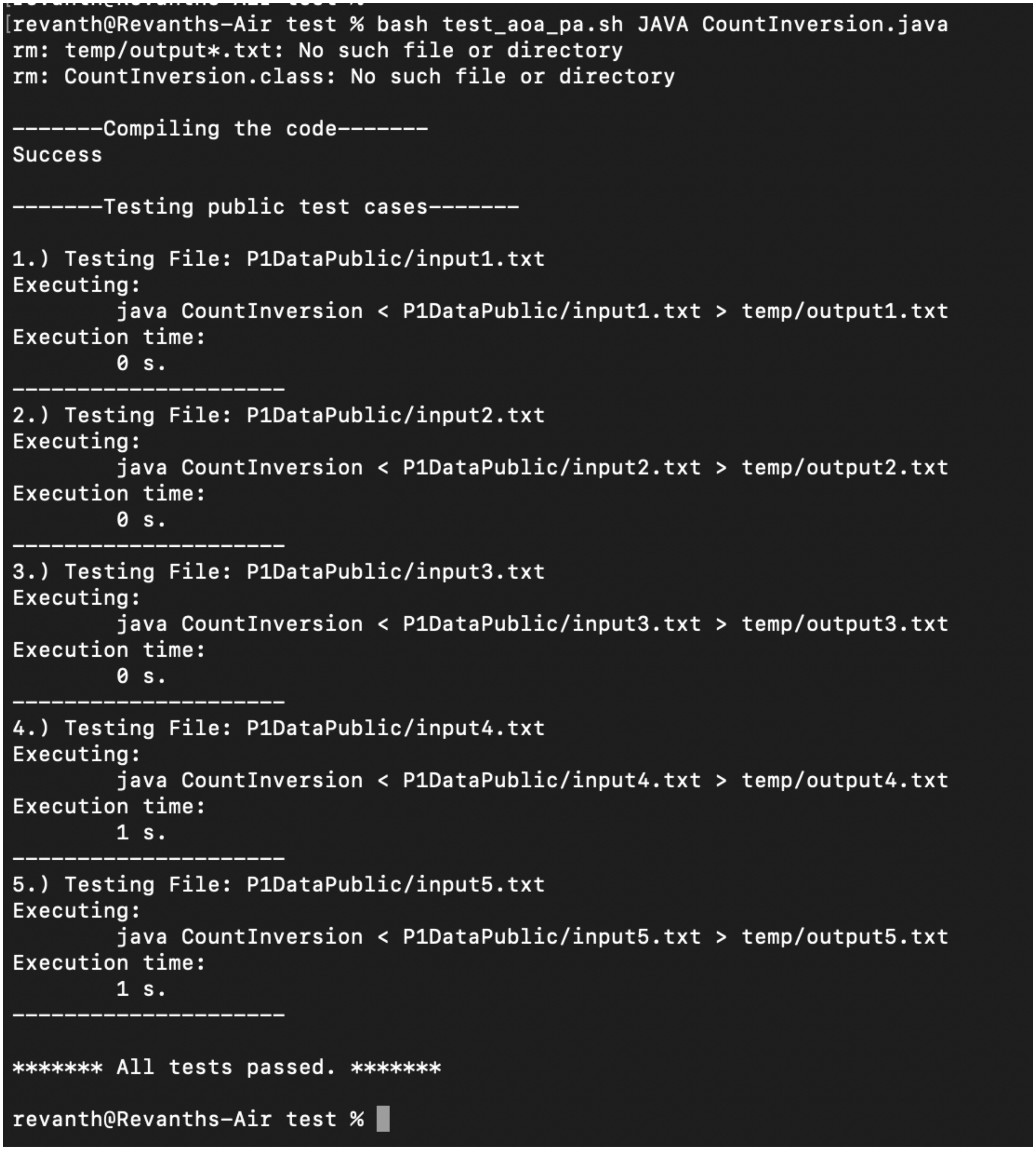
   C++
   Compile: g++ -std=c++11 -o <executable_name> <filename>
   Execution: ./<executable_name>

   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

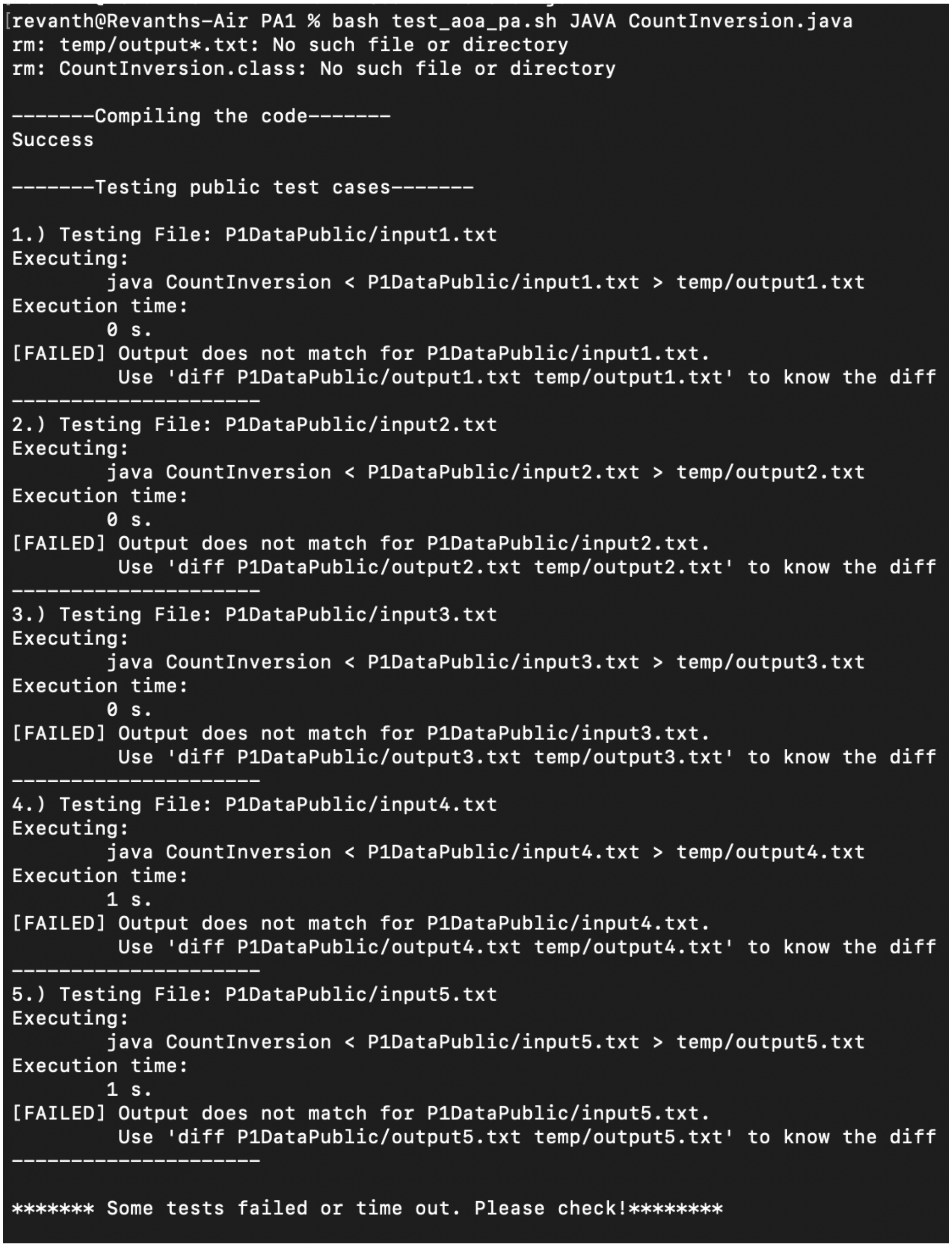
If you get below output, your input/output processing is correct.



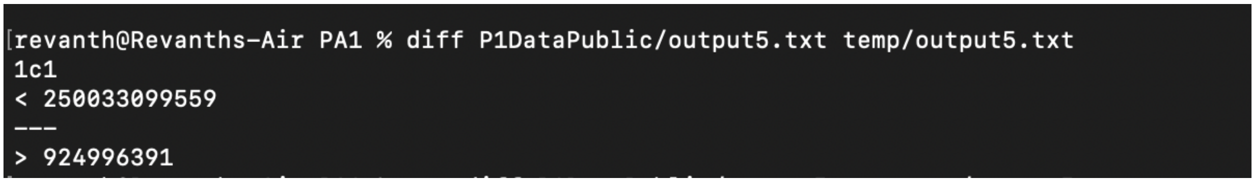
```
[revanth@Revanths-Air test % bash test_aoa_pa.sh JAVA CountInversion.java
rm: temp/output*.txt: No such file or directory
rm: CountInversion.class: No such file or directory

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

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

1.) Testing File: P1DataPublic/input1.txt
Executing:
        java CountInversion < P1DataPublic/input1.txt > temp/output1.txt
Execution time:
        0 s.
---------------------
2.) Testing File: P1DataPublic/input2.txt
Executing:
        java CountInversion < P1DataPublic/input2.txt > temp/output2.txt
Execution time:
        0 s.
---------------------
3.) Testing File: P1DataPublic/input3.txt
Executing:
        java CountInversion < P1DataPublic/input3.txt > temp/output3.txt
Execution time:
        0 s.
---------------------
4.) Testing File: P1DataPublic/input4.txt
Executing:
        java CountInversion < P1DataPublic/input4.txt > temp/output4.txt
Execution time:
        1 s.
---------------------
5.) Testing File: P1DataPublic/input5.txt
Executing:
        java CountInversion < P1DataPublic/input5.txt > temp/output5.txt
Execution time:
        1 s.
---------------------

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

revanth@Revanths-Air test %
```

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

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



```
[revanth@Revanths-Air PA1 % bash test_aoa_pa.sh JAVA CountInversion.java
rm: temp/output*.txt: No such file or directory
rm: CountInversion.class: No such file or directory

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

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

1.) Testing File: P1DataPublic/input1.txt
Executing:
        java CountInversion < P1DataPublic/input1.txt > temp/output1.txt
Execution time:
        0 s.
[FAILED] Output does not match for P1DataPublic/input1.txt.
         Use 'diff P1DataPublic/output1.txt temp/output1.txt' to know the diff
---------------------
2.) Testing File: P1DataPublic/input2.txt
Executing:
        java CountInversion < P1DataPublic/input2.txt > temp/output2.txt
Execution time:
        0 s.
[FAILED] Output does not match for P1DataPublic/input2.txt.
         Use 'diff P1DataPublic/output2.txt temp/output2.txt' to know the diff
---------------------
3.) Testing File: P1DataPublic/input3.txt
Executing:
        java CountInversion < P1DataPublic/input3.txt > temp/output3.txt
Execution time:
        0 s.
[FAILED] Output does not match for P1DataPublic/input3.txt.
         Use 'diff P1DataPublic/output3.txt temp/output3.txt' to know the diff
---------------------
4.) Testing File: P1DataPublic/input4.txt
Executing:
        java CountInversion < P1DataPublic/input4.txt > temp/output4.txt
Execution time:
        1 s.
[FAILED] Output does not match for P1DataPublic/input4.txt.
         Use 'diff P1DataPublic/output4.txt temp/output4.txt' to know the diff
---------------------
5.) Testing File: P1DataPublic/input5.txt
Executing:
        java CountInversion < P1DataPublic/input5.txt > temp/output5.txt
Execution time:
        1 s.
[FAILED] Output does not match for P1DataPublic/input5.txt.
         Use 'diff P1DataPublic/output5.txt temp/output5.txt' to know the diff
---------------------

******* Some tests failed or time out. Please check!********
```



```
[revanth@Revanths-Air PA1 % diff P1DataPublic/output5.txt temp/output5.txt
1c1
< 250033099559
---
> 924996391
```

Here in this diff command, the first line is expected and the second line is actual.

The script should be complete in a few seconds. If the script hangs, then your input/output processing is wrong.