Brendan Henderson
Christopher Tang
OMIS107

autoGrader.sh

Brendan and Christopher designed a directory within the bdhenderson directory that holds relevant subdirectories for the script program, autoGrader.sh. This directory consists of the main two subdirectories, Students and the Instructor.

Brendan and Christopher began the project by using the website *Lucidchart* to help visualize the task of creating the entire project directory based on the guidelines of the assignment. The Lucidflow chart is not 100% accurate to the completed project that Brendan and Christopher turned in, as there were alternations made during the creation process for various reasons. However, the Lucidchart planning phase allowed us to have a jumping-off point for this project.

The main focus of this project is on HW3 (homework3). Most other files in this directory that do not specifically pertain to HW3 exist as fillers that can act as an added layer of detail to our student grading system.

We have three students within our finalProject Students directory that are directly called upon within our autoGrader script. The script grades each of the three students for each of the three problems within HW3. The script compares each of the students' code output against that of the Instructor's, making a total of 9 grades.

This project was largely handwritten, for the sake of transparency and the professor's interest in how students have been using generative AI, we used sources like ChatGPT to help with a couple of specific instances within the assignment, and quite often for debugging purposes if we became stuck on something, or could not spot an error ourselves.

The script code is long and is likely not the most efficient method of achieving our goals, however, for the sake of attempting to complete the project largely without external tools help, it came at the cost of code efficiency, which was great practice.

Most of this documented code was written after the creation of the program. Please forgive us if there are a few small errors within this documented code, as for some of the file directory creations we did not log all our code alongside the actual creation of it, however, we tried our best to rewrite all code for this program's build.

We have attached the presentation link as well as the project tree and Lucid chart link at the end of this document.

Lastly, for the sake of simplicity, we will assume that all locations are stored within: root/home/bdhenderson/ for the rest of this code documentation and referencing below.

Initial Setup:

Code:
mkdir finalProject

Functionality:
Makes final project directory within student's personal directory

Copied into finalProject alice.txt

Code:
cp /home/OMIS107/Lecture2/alice.txt /home/bdhenderson/finalProject

Functionality:
copied alice.txt into finalProject

Creation of Students Directory

Code:
mkdir /finalProject/Students

Functionality:
Makes Students directory within finalProject directory

students(1-3) directories:

Code:
mkdir student1 student2 studen3 /finalProject/Students

Functionality:
Made three default student directory accessible to the student

---

Code:

```
mkdir HW1 HW2 HW3  /finalProject/Students/student1
mkdir HW1 HW2 HW3  /finalProject/Students/student2
mkdir HW1 HW2 HW3  /finalProject/Students/student3
```

Functionality:

Makes 3 HW directories for each of the three students

Code:

```
cp ../../alice.txt /finalProject/Students/student1
cp ../../alice.txt /finalProject/Students/student2
cp ../../alice.txt /finalProject/Students/student3
```

Functionality:

copied alice.txt from finalProject dir into each student file

---

Code:

```
touch /finalProject/Students/student1/HW1/HW1problems.csv
touch /finalProject/Students/student1/HW2/HW2problems.csv

touch /finalProject/Students/student2/HW1/HW1problems.csv
touch /finalProject/Students/student2/HW2/HW2problems.csv

touch /finalProject/Students/student3/HW1/HW1problems.csv
touch /finalProject/Students/student3/HW2/HW2problems.csv
```

Functionality:

Creates empty tables called HW1problems and HW2problems. These are empty and are the same as HW3questions(not empty) however it was used to distinguish how they won't be used in auto grader (they are filler).

Code:

```
touch /finalProject/Students/student1/HW3/HW3questions.csv
touch /finalProject/Students/student2/HW3/HW3questions.csv
touch /finalProject/Students/student3/HW3/HW3questions.csv
```

Functionality:

Creates HW3questions empty table for HW3 within each student directory.

Code:

```
printf “Problem\t\Question\n” > /finalProject/Students/student1/HW3/HW3questions.csv
printf “Problem\t\Question\n” > /finalProject/Students/student2/HW3/HW3questions.csv
printf “Problem\t\Question\n” > /finalProject/Students/student3/HW3/HW3questions.csv
```

Functionality:

Creates HW3questions.csv header for each student.

Code:

```
printf “1\t\Output all lines in alice.txt that contain the word hold\n” >>
/finalProject/Students/student1/HW3/HW3questions.csv
printf “1\t\Output all lines in alice.txt that contain the word hold\n” >>
/finalProject/Students/student2/HW3/HW3questions.csv
printf “1\t\Output all lines in alice.txt that contain the word hold\n” >>
/finalProject/Students/student3/HW3/HW3questions.csv

printf “2\t\Output all lines in alice.txt that contain the word hare surrounded by spaces\n” >>
/finalProject/Students/student1/HW3/HW3questions.csv
printf “2\t\Output all lines in alice.txt that contain the word hare surrounded by spaces\n” >>
/finalProject/Students/student2/HW3/HW3questions.csv
printf “2\t\Output all lines in alice.txt that contain the word hare surrounded by spaces\n” >>
/finalProject/Students/student3/HW3/HW3questions.csv

printf “3\t\Output all lines in alice.txt that contain the word it immediately followed by a period\n” >>
/finalProject/Students/student1/HW3/HW3questions.csv
printf “3\t\Output all lines in alice.txt that contain the word it immediately followed by a period\n” >>
/finalProject/Students/student1/HW3/HW3questions.csv
printf “3\t\Output all lines in alice.txt that contain the word it immediately followed by a period\n”
/finalProject/Students/student1/HW3/HW3questions.csv
```

Functionality:

Fills in HW3questions.csv for each student and each question

Code:

```
touch /finalProject/Students/student1/HW1/HW1submission.csv
touch /finalProject/Students/student2/HW1/HW1submission.csv
touch /finalProject/Students/student3/HW1/HW1submission.csv
```

Functionality:

Creates HW1submission.csv empty table for each student (not used in the final version), only for looks

---

Code:

```
touch /finalProject/Students/student1/HW2/HW2submission.csv
touch /finalProject/Students/student2/HW2/HW2submission.csv
touch /finalProject/Students/student3/HW2/HW2submission.csv
```

Functionality:

Creates HW2submission.csv empty table for each student (not used in the final version), only for looks

---

Code:

```
 touch /finalProject/Students/student1/HW3/HW3submission.csv
 touch /finalProject/Students/student2/HW3/HW3submission.csv
 touch /finalProject/Students/student3/HW3/HW3submission.csv
```

Functionality:

Creates HW3submission.csv empty table for each student for HW3 to be picked up by autoGrade

---

Code:

```
printf “Problem\tSubmission\n” > /finalProject/Students/student1/HW3/HW3submission.csv
printf “Problem\tSubmission\n” > /finalProject/Students/student2/HW3/HW3submission.csv
printf “Problem\tSubmission\n” > /finalProject/Students/student3/HW3/HW3submission.csv
```

Functionality:

Makes headers for each HW3submission.csv table

Code:

```
printf “1\t grep -Ein \" hold \" ../alice.txt\n” >> /finalProject/Students/student1/HW3/HW3submission.csv
printf “2\t grep -Ein \" hare \" ../alice.txt\n” >> /finalProject/Students/student1/HW3/HW3submission.csv
printf “3\t grep -Ein \" it\. \" ../alice.txt\n” >> /finalProject/Students/student1/HW3/HW3submission.csv
```

Functionality:

Inserts student1 submission for HW3 to be read by autograder script.

Code:

```
printf “1\tgrep -En \"hold \" ../alice.txt\n” >> /finalProject/Students/student2/HW3/HW3submission.csv
printf “2\tgrep -Ein \" hare \" ../alice.txt\n” >> /finalProject/Students/student2/HW3/HW3submission.csv
printf “3\tgrep -Ein \" it\.\" ../alice.txt\n” >> /finalProject/Students/student2/HW3/HW3submission.csv
```

Functionality:
Inserts student2 submission for HW3 to be read by autograder script.

Code:
printf “1\tgrep -En \" hold \" ../alice.txt” >> /finalProject/Students/student3/HW3/HW3submission.csv
printf “2\tgrep -Ein \" hare \" ../alice.txt” >> /finalProject/Students/student3/HW3/HW3submission.csv
printf “3\tgrep -Ein \"it. \" ../alice.txt” >>
/finalProject/Students/student3/HW3/HW3submission.csv

Functionality:
Inserts student3 submission for HW3 to be read by autograder script

---

Creation of Instructor Directory

Code:
mkdir /finalProject/Instructor

Functionality:
Makes subdirectory Instructor within finalProject directory.

Instructor/homework

Code:
mkdir /finalProject/Instructor/homework

Functionality:
Makes subdirectory homework within Instructor directory

Instructor/homework/questions

Code:
mkdir finalProject/Instructor/homework/questions

Functionality:
Makes subdirectory of subdirectory homework within Instructor for assigned student homework questions.

Code:
touch /finalProject/Instructor/questions/HW1questions.csv
touch /finalProject/Instructor/questions/HW2questions.csv
touch /finalProject/Instructor/questions/HW3questions.csv

Functionality:
Creates empty tables, csv, for each HW within the Instructor's directory questions

---

Code:
printf "Problem\tQuestion\n" > /finalProject/Instructor/questions/HW1questions.csv
printf "Problem\tQuestion\n" > /finalProject/Instructor/questions/HW2questions.csv
printf "Problem\tQuestion\n" > /finalProject/Instructor/questions/HW3questions.csv

Functionality:
Creates instructor questions table headers

Code:
printf "1\tOutput all lines in alice.txt that contain the word hold\n" >> HW3questions.csv
printf "2\tOutput all lines in alice.txt that contain the word "hare" surrounded by a space on both ends\n" >> HW3questions.csv
printf "3\tOutput all lines in alice.txt that contain the word it that is immediately followed by a period\n" >> HW3questions.csv

Functionality:
Provides questions for HW3, the HW to be graded by autograder.

Instructor/homework/solutions

Code:

mkdir finalProject/Instructor/homework/solutions

Functionality:
Makes subdirectory of subdirectory homework within Instructor for assigned student homework solutions.

Code:
touch /finalProject/Instructor/solutions/HW1solutions.csv
touch /finalProject/Instructor/solutions/HW2solutions.csv
touch /finalProject/Instructor/solutions/HW3solutions.csv

Functionality: **Makes three empty tables within solutions folder for each specific HW solution**

Code:
printf "Problem\tSolution\n" > HW1solutions.csv
printf "Problem\tSolution\n" > HW2solutions.csv
printf "Problem\tSolution\n" > HW3solutions.csv

Functionality:
Declared default header for instructor HW solutions

Code:
printf “1\t grep -Ein \" hold \" ../alice.txt\n >> HW3solutions.csv
printf “2\t grep -Ein \" hare \" ../alice.txt\n >> HW3solutions.csv
printf “3\t grep -Ein \" it\. \" ../alice.txt\n >> HW3solutions.csv

Functionality:
Fills out fields in HW3solutions, the solutions that are used for autograder to work

---

Instructor/homework/expectedOutputs

Code:
mkdir /finalProject/Instructor/homework/expectedOutputs

Functionality:

Makes subdirectory of subdirectory homework within Instructor for student code expected outputs.

Code:
touch /finalProject/Instructor/expectedOutputs/HW1expected.csv
touch /finalProject/Instructor/expectedOutputs/HW2expected.csv
touch /finalProject/Instructor/expectedOutputs/HW3expected.csv

Functionality:
Creates three empty tables within expectedoutput, one for each HW

Code:
printf "Problem\tOutput\n" > HW1expected.csv
printf "Problem\tOutput\n" > HW2expected.csv
printf "Problem\tOutput\n" > HW3expected.csv

Functionality:
Makes headers for expected tables, all tables are empty, except the table HW3 output because it is calculated within the script

---

Instructor/mainTable.csv

Code:
touch /finalProject/Instuctor/mainTable.csv

Functionality:
Creates empty main table for autograde referencing.

Code:
printf "Student\tHW1\tHW2\n” > /finalProject/Instuctor/mainTable.csv

Functionality:
Creates basic header structure for mainTable.csv

Code:

```
printf "student1\t30\t25\n" >> mainTable.csv
printf "student2\t20\t15\n" >> mainTable.csv
printf “student3\t10\t10\n” >> mainTable.csv
```

Functionality:

Creates basic default students and student grades for non-autograde affected code(filler).

Instructor/roster.csv

Code:

```
touch /finalProject/Instructor/roster.csv
```

Functionality:

Creates empty csv table for the roster.

Code:

```
printf "Student\n" > /finalProject/Instructor/roster.csv
```

Functionality:

Creates field headers for roster.csv using default students and grades

Code:

```
printf "John\n" >> /finalProject/Instructor/roster.csv
printf "Becky\n" >> /finalProject/Instructor/roster.csv
printf "Riley\n" >> /finalProject/Instructor/roster.csv
```

Functionality:

Creates rows of student names within roster.csv to be referred to in autograde.

Instructor/autoGrader.sh

Code:
vi autoGrader.sh

Functionality:
creates script for autoGrader.sh

Code:
chmod g+rwx myscript.sh

Functionality:
Made autoGrader script read write executable for anyone within OMIS107 group

autoGrader.sh

```
#!/bin/bash

# Extract the command from the second line, second field

#------------INSTRUCTOR CODE--------------------------------------------------------------------------


#fetch Instructor homework 3 solutions and store in respective solution variables below:

hw3p1_solution=$(awk -F'\t' '{if (NR==2) print $2}' homework/solutions/HW3solutions.csv)
hw3p2_solution=$(awk -F'\t' '{if (NR==3) print $2}' homework/solutions/HW3solutions.csv)
hw3p3_solution=$(awk -F'\t' '{if (NR==4) print $2}' homework/solutions/HW3solutions.csv)

#run each of the Instructor's homework3 problem solutions, and store each in variables below
hw3p1_expOutput=$(eval $hw3p1_solution)
hw3p2_expOutput=$(eval $hw3p2_solution)
hw3p3_expOutput=$(eval $hw3p3_solution)
```

```
#------------MISC CODE-------------------------------------------------------------------------


#retrives actual roster names and assigns them to student default names
student1name=$(awk 'NR==2 {print $0}' roster.csv)

student2name=$(awk 'NR==3 {print $0}' roster.csv)

student3name=$(awk 'NR==4 {print $0}' roster.csv)







#intro timed initiator

echo
echo
echo
echo
echo
-------------------------------------------------------------------------------------------------------------------
-------------
echo
_____________________________________________________AUTOGRADER_________
_____________________________________________________
echo
-------------------------------------------------------------------------------------------------------------------
-------------
echo
echo
echo
echo
echo
echo
sleep 1
```

```
echo .
sleep 1
echo ..
sleep 1
echo ...
echo
echo Calculating student1 grades
echo
sleep 2



#------------STUDENT CODE-------------------------------------------------------------------------




#Student1-----------------------------------------------------------------------------------------------------

#initialize studen1 HW3 grade
student1Grade=0

#fetch student1 homework3 submissions and store in variables stu1hw3p1_submission,
stu1hw3p2_submission, stu1hw3p3_submission
stu1hw3p1_submission=$(awk -F'\t' '{if (NR==2) print $2}'
../Students/student1/HW3/HW3submission.csv)
stu1hw3p2_submission=$(awk -F'\t' '{if (NR==3) print $2}'
../Students/student1/HW3/HW3submission.csv)
stu1hw3p3_submission=$(awk -F'\t' '{if (NR==4) print $2}'
../Students/student1/HW3/HW3submission.csv)



#run each of student1 homework3 problem submissions, and respectively store each in
stu1hw3p1_output, stu1hw3p2_output,stu1hw3p3_output
stu1hw3p1_output=$(eval $stu1hw3p1_submission)
stu1hw3p2_output=$(eval $stu1hw3p2_submission)
stu1hw3p3_output=$(eval $stu1hw3p3_submission)


echo ----------------------------------------------------
```

```
echo $student1name - Results:
echo ------------------------------------------------


#----------STUDENT 1 HOMEWORK3 PROBLEM 1 DIFFERENCES CALC
----------------------------------------

#checks if student3's homework 3 problem 1 submission matches Instructor's homework 3 problem 1
solution
hw3_p1_differences=$(diff <(echo "$stu1hw3p1_output") <(echo "$hw3p1_expOutput") | wc -l)


#if there is a 0 line difference, student gets 10/10 pts for that problem
if [ "$hw3_p1_differences" -eq 0 ]; then
        no_loss=10
        student1Grade=$((student1Grade + 10))
        echo "student1 recieves $no_loss out of 10 for HW3 Problem 1 "


#if there is a 1-9 line difference, student gets 1-9/10 pts for that problem
elif [ "$hw3_p1_differences" -le 10 ]; then
        point_loss=$((hw3_p1_differences*1))
        totalploss=$((10 - point_loss))
        student1Grade=$((student1Grade + totalploss))
        echo "student1 recieves $totalploss out of 10 for HW3 Problem 1 "

#if there is a 10+ line difference, student gets 0/10 pts for that problem

else
        all_loss=0
        student1Grade=$((student1Grade + 0))
        echo "student1 recieves $all_loss out of 10 for HW3 Problem 1 "

fi

#----------STUDENT 1 HOMEWORK3 PROBLEM 2 DIFFERENCES CALC
----------------------------------------

hw3_p2_differences=$(diff <(echo "$stu1hw3p2_output") <(echo "$hw3p2_expOutput") | wc -l)

#if there is a 0 line difference, student gets 10/10 pts for that problem
if [ "$hw3_p2_differences" -eq 0 ]; then
        no_loss=10
        student1Grade=$((student1Grade + 10))
```

```
        echo "student1 recieves $no_loss out of 10 for HW3 Problem 2 "


#if there is a 1-9 line difference, student gets 1-9/10 pts for that problem
elif [ "$hw3_p2_differences" -le 10 ]; then
        point_loss=$((hw3_p2_differences*1))
        totalploss=$((10 - point_loss))
        student1Grade=$((student1Grade + totalploss))


        echo "student1 recieves $totalploss out of 10 for HW3 Problem 2 "

#if there is a 10+ line difference, student gets 0/10 pts for that problem

else
        all_loss=0
        student1Grade=$((student1Grade + 0))
        echo "student1 recieves $all_loss out of 10 for HW3 Problem 2 "

fi

#---------------STUDENT 1 HOMEWORK3 PROBLEM 3 DIFFERENCES CALC
-----------------------------------------

hw3_p3_differences=$(diff <(echo "$stu1hw3p3_output") <(echo "$hw3p3_expOutput") | wc -l)


#if there is a 0 line difference, student gets 10/10 pts for that problem
if [ "$hw3_p3_differences" -eq 0 ]; then
        no_loss=10
        student1Grade=$((student1Grade + 10))
        echo "student1 recieves $no_loss out of 10 for HW3 Problem 3 "


#if there is a 1-9 line difference, student gets 1-9/10 pts for that problem
elif [ "$hw3_p3_differences" -le 10 ]; then
        point_loss=$((hw3_p3_differences*1))
        totalploss=$((10 - point_loss))
        student1Grade=$((student1Grade + totalploss))

        echo "student1 recieves $totalploss out of 10 for HW3 Problem 3 "

#if there is a 10+ line difference, student gets 0/10 pts for that problem
```

```
else
	all_loss=0
	student1Grade=$((student1Grade + 0))
	echo "student1 recieves $all_loss out of 10 for HW3 Problem 3 "

fi

echo
echo student1 total HW3 grade: $student1Grade out of 30



#Student2-------------------------------------------------------------------------
echo
sleep 1
echo Calculating student2 grades
sleep 2
echo .
sleep 1
echo ..
sleep 1
echo ...
echo



echo ------------------------------------------------------
echo $student2name - Results:
echo ------------------------------------------------------

stu2hw3p1_submission=$(awk -F'\t' '{if (NR==2) print $2}'
../Students/student2/HW3/HW3submission.csv)
stu2hw3p2_submission=$(awk -F'\t' '{if (NR==3) print $2}'
../Students/student2/HW3/HW3submission.csv)
stu2hw3p3_submission=$(awk -F'\t' '{if (NR==4) print $2}'
../Students/student2/HW3/HW3submission.csv)

stu2hw3p1_output=$(eval $stu2hw3p1_submission)
stu2hw3p2_output=$(eval $stu2hw3p2_submission)
stu2hw3p3_output=$(eval $stu2hw3p3_submission)

#----------STUDENT 2 HOMEWORK3 PROBLEM 1 DIFFERENCES CALC
----------------------------------------
```

```
hw3_p1_differences=$(diff <(echo "$stu2hw3p1_output") <(echo "$hw3p1_expOutput") | wc -l)

student2Grade=0

#if there is a 0 line difference, student gets 10/10 pts for that problem
if [ "$hw3_p1_differences" -eq 0 ]; then
	no_loss=10
	student2Grade=$((student2Grade + 10))
	echo "student2 recieves $no_loss out of 10 for HW3 Problem 1 "


#if there is a 1-9 line difference, student gets 1-9/10 pts for that problem
elif [ "$hw3_p1_differences" -le 10 ]; then
	point_loss=$((hw3_p1_differences*1))
	totalploss=$((10 - point_loss))
	student2Grade=$((student2Grade + totalploss))
	echo "student2 recieves $totalploss out of 10 for HW3 Problem 1 "

#if there is a 10+ line difference, student gets 0/10 pts for that problem

else
	all_loss=0
	student2Grade=$((student2Grade + 0))
	echo "student2 recieves $all_loss out of 10 for HW3 Problem 1 "

fi

#----------------STUDENT 2 HOMEWORK3 PROBLEM 2 DIFFERENCES CALC
-----------------------------------------

hw3_p2_differences=$(diff <(echo "$stu2hw3p2_output") <(echo "$hw3p2_expOutput") | wc -l)

#if there is a 0 line difference, student gets 10/10 pts for that problem
if [ "$hw3_p2_differences" -eq 0 ]; then
	no_loss=10
	student2Grade=$((student2Grade + 10))

	echo "student2 recieves $no_loss out of 10 for HW3 Problem 2 "


#if there is a 1-9 line difference, student gets 1-9/10 pts for that problem
elif [ "$hw3_p2_differences" -le 10 ]; then
	point_loss=$((hw3_p2_differences*1))
	totalploss=$((10 - point_loss))
```

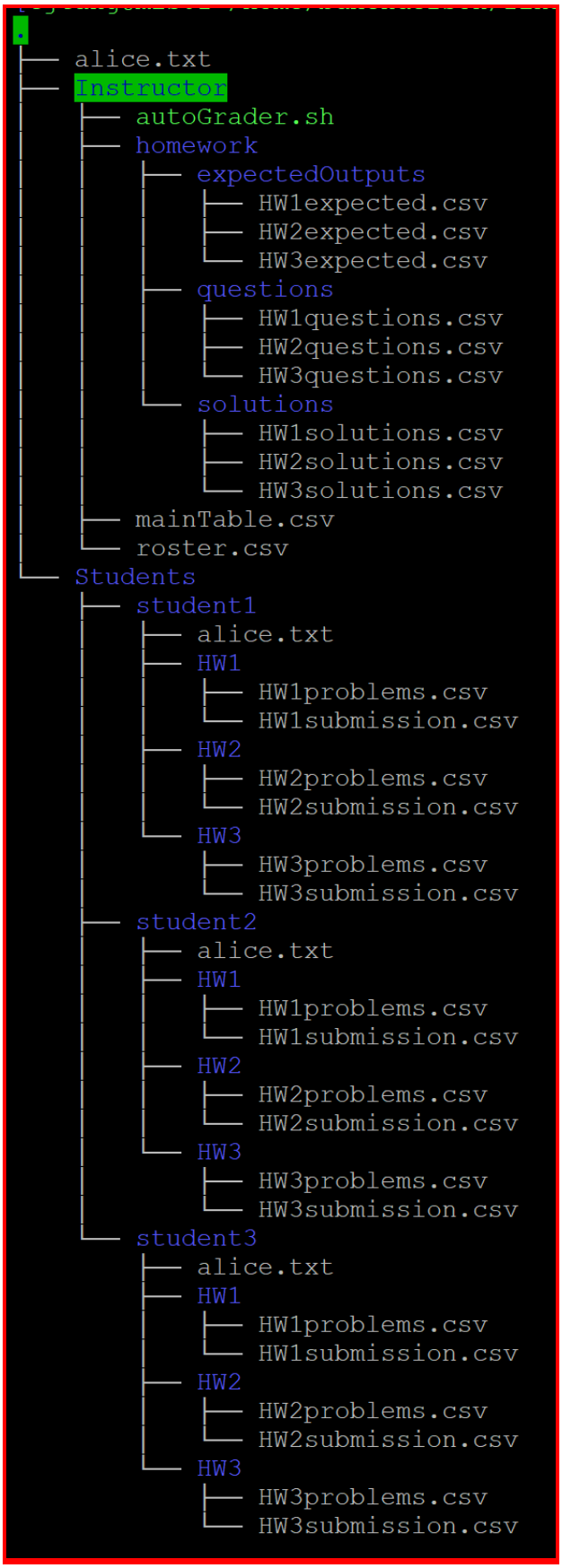
```
        student2Grade=$((student2Grade + totalploss))
        echo "student2 recieves $totalploss out of 10 for HW3 Problem 2 "

#if there is a 10+ line difference, student gets 0/10 pts for that problem

else
        all_loss=0
        student2Grade=$((student2Grade + 0))
        echo "student2 recieves $all_loss out of 10 for HW3 Problem 2 "

fi

#---------------STUDENT 2 HOMEWORK3 PROBLEM 3 DIFFERENCES CALC
----------------------------------------

hw3_p3_differences=$(diff <(echo "$stu2hw3p3_output") <(echo "$hw3p3_expOutput") | wc -l)

#if there is a 0 line difference, student gets 10/10 pts for that problem
if [ "$hw3_p3_differences" -eq 0 ]; then
        no_loss=10
        student2Grade=$((student2Grade + 10))

        echo "student2 recieves $no_loss out of 10 for HW3 Problem 3 "


#if there is a 1-9 line difference, student gets 1-9/10 pts for that problem
elif [ "$hw3_p3_differences" -le 10 ]; then
        point_loss=$((hw3_p3_differences*1))
        totalploss=$((10 - point_loss))
        student2Grade=$((student2Grade + totalploss))
        echo "student2 recieves $totalploss out of 10 for HW3 Problem 3 "

#if there is a 10+ line difference, student gets 0/10 pts for that problem

else
        all_loss=0
        student2Grade=$((student2Grade + 0))
        echo "student2 recieves $all_loss out of 10 for HW3 Problem 3 "

fi

echo
echo student2 total HW3 grade: $student2Grade out of 30
echo
```

```
sleep 1



#Student3-------------------------------------------------------------------------------


echo Calculating student3 grades
sleep 2
echo .
sleep 1
echo ..
sleep 1
echo ...
echo
echo ----------------------------------------------------
echo $student3name - Results:
echo ----------------------------------------------------

stu3hw3p1_submission=$(awk -F'\t' '{if (NR==2) print $2}'
../Students/student3/HW3/HW3submission.csv)
stu3hw3p2_submission=$(awk -F'\t' '{if (NR==3) print $2}'
../Students/student3/HW3/HW3submission.csv)
stu3hw3p3_submission=$(awk -F'\t' '{if (NR==4) print $2}'
../Students/student3/HW3/HW3submission.csv)

stu3hw3p1_output=$(eval $stu3hw3p1_submission)
stu3hw3p2_output=$(eval $stu3hw3p2_submission)
stu3hw3p3_output=$(eval $stu3hw3p3_submission)

#---------------STUDENT 3 HOMEWORK3 PROBLEM 1 DIFFERENCES CALC
----------------------------------------

hw3_p1_differences=$(diff <(echo "$stu3hw3p1_output") <(echo "$hw3p1_expOutput") | wc -l)


student3Grade=0
#if there is a 0 line difference, student gets 10/10 pts for that problem
if [ "$hw3_p1_differences" -eq 0 ]; then
        no_loss=10
        student3Grade=$((student3Grade + 10))
        echo "student3 recieves $no_loss out of 10 for HW3 Problem 1 "
```

```
#if there is a 1-9 line difference, student gets 1-9/10 pts for that problem
elif [ "$hw3_p1_differences" -le 10 ]; then
	point_loss=$((hw3_p1_differences*1))
	totalploss=$((10 - point_loss))
	student3Grade=$((student3Grade + totalploss))
	echo "student3 recieves $totalploss out of 10 for HW3 Problem 1 "

#if there is a 10+ line difference, student gets 0/10 pts for that problem

else
	all_loss=0
	student3Grade=$((student3Grade + 0))
	echo "student3 recieves $all_loss out of 10 for HW3 Problem 1 "

fi

#----------STUDENT 3 HOMEWORK3 PROBLEM 2 DIFFERENCES CALC
----------

hw3_p2_differences=$(diff <(echo "$stu3hw3p2_output") <(echo "$hw3p2_expOutput") | wc -l)


#if there is a 0 line difference, student gets 10/10 pts for that problem
if [ "$hw3_p2_differences" -eq 0 ]; then
	no_loss=10
	student3Grade=$((student3Grade + 10))
	echo "student3 recieves $no_loss out of 10 for HW3 Problem 2 "


#if there is a 1-9 line difference, student gets 1-9/10 pts for that problem
elif [ "$hw3_p2_differences" -le 10 ]; then
	point_loss=$((hw3_p2_differences*1))
	totalploss=$((10 - point_loss))

	student3Grade=$((student3Grade + totalploss))
	echo "student3 recieves $totalploss out of 10 for HW3 Problem 2 "

#if there is a 10+ line difference, student gets 0/10 pts for that problem

else
	all_loss=0
	student3Grade=$((student3Grade + 0))
	echo "student3 recieves $all_loss out of 10 for HW3 Problem 2 "
```

```
fi

#---------------STUDENT 3 HOMEWORK3 PROBLEM 3 DIFFERENCES CALC ---------------------------

hw3_p3_differences=$(diff <(echo "$stu3hw3p3_output") <(echo "$hw3p3_expOutput") | wc -l)

#if there is a 0 line difference, student gets 10/10 pts for that problem
if [ "$hw3_p3_differences" -eq 0 ]; then
        no_loss=10
        student3Grade=$((student3Grade + 10))
        echo "student3 recieves $no_loss out of 10 for HW3 Problem 3 "


#if there is a 1-9 line difference, student gets 1-9/10 pts for that problem
elif [ "$hw3_p3_differences" -le 10 ]; then
        point_loss=$((hw3_p3_differences*1))
        totalploss=$((10 - point_loss))
        student3Grade=$((student3Grade + totalploss))

        echo "student3 recieves $totalploss out of 10 for HW3 Problem 3 "

#if there is a 10+ line difference, student gets 0/10 pts for that problem

else
        all_loss=0
        student3Grade=$((student3Grade + 0))
        echo "student3 recieves $all_loss out of 10 for HW3 Problem 3 "

fi

echo
echo student3 total HW3 grade: $student3Grade out of 30




#---ENDING
CODE----------------------------------------------------------------------------------------------------
echo
sleep 2

# Create new_column.csv with HW3 grades
echo "HW3" > new_column.csv
echo "$student1Grade" >> new_column.csv
```

```
echo "$student2Grade" >> new_column.csv
echo "$student3Grade" >> new_column.csv

echo temp file created!

sleep 1
echo


paste -d $'\t' mainTable.csv new_column.csv > updatedMainTable.csv



echo new updatedMainTable.csv created!
sleep 1
echo
echo
echo
#prints out properly aligned updatedMainTable instead of just cat
echo Updated HW Grade Table
echo ----------------------
column -t -s $'\t' updatedMainTable.csv
echo ----------------------
echo
echo

#removes new column and updatedMainTable.csv from Instructor
dir-----------------------------------------------------

rm new_column.csv
sleep 1
echo temp column removed from Instructor dir!

rm updatedMainTable.csv
sleep 1
echo updatedMainTable removed from Instructor dir!
echo
```


.
alice.txt
Instructor
autoGrader.sh
homework
expectedOutputs
HW1expected.csv
HW2expected.csv
HW3expected.csv
questions
HW1questions.csv
HW2questions.csv
HW3questions.csv
solutions
HW1solutions.csv
HW2solutions.csv
HW3solutions.csv
mainTable.csv
roster.csv
Students
student1
alice.txt
HW1
HW1problems.csv
HW1submission.csv
HW2
HW2problems.csv
HW2submission.csv
HW3
HW3problems.csv
HW3submission.csv
student2
alice.txt
HW1
HW1problems.csv
HW1submission.csv
HW2
HW2problems.csv
HW2submission.csv
HW3
HW3problems.csv
HW3submission.csv
student3
alice.txt
HW1
HW1problems.csv
HW1submission.csv
HW2
HW2problems.csv
HW2submission.csv
HW3
HW3problems.csv
HW3submission.csv

Presentation:
https://docs.google.com/presentation/d/1qCj9hm3oMsmY1SXrlzVctvkCnP_W-q7cX_G9jW-m2tg/edit?usp=sharing

Lucidchart:
https://lucid.app/lucidchart/94cd36c2-7660-468d-8149-3d8495d3296e/edit?viewport_loc=-4142%2C-1808%2C10500%2C6175%2Csebq8KMfXKJJH&invitationId=inv_77e467be-4ea2-49a3-9ff3-161b29329447