Liam McElroy
Austin Dickieson

Automatic Grader

Defining Variables:

Our script relies on a few conditions, most of which pertain to file names and directory organization.
We assume students have directories under the same name as the roster and that within the direction is a file named *homework#.txt* for any passed in #.
The student homeworks must have the same number of lines as the instructor solution, which is stored in his/her directory under *fakesolution#.txt*
These lines must contain code to be run and compared to the instructors output
There also must be a pre-existing master table for which to paste the calculated scores for each homework. This table holds the roster and appends each homework with every students score as it is graded.
Attached below is a visual of the directories before and after the program runs

Instructor Loop

```
echo Enter the homework number
read hw_number
number_of_problems=$(cat instructor/fakesolution$hw_number.txt | grep -E "." | wc -l)
echo "HW_$hw_number" > HW_Scores_$hw_number
```

For each homework problem, create a file containing the correct solution

(1) Find *number_of_problems* for which to loop through the instructor solution and eventually compare student to instruction.

(2) The instructor loop below then goes through each line of *fakesolutions#.txt* and writes output of that code to a file titled *hw_solution_#*

```
for problem in $(seq $number_of_problems)
do
    output=$(cat instructor/fakesolution$hw_number.txt | grep -E "." | head -$problem | tail -1)
    eval $output > instructor/hw_solution_$problem
done
```

Student Loop

Loop through each student in *fakeroster.txt* and declare *incorrect* (the number of incorrect problems for each student) to be 0

```
for student in $(cat fakeroster.txt)
do
    incorrect=0
```

Problem Loop (nested within the Student Loop)

For each student, loop through each problem in in *number_of_problems* to:

(1) define the student's homework file as *sfile*

(2) create a file for that contains the student's code for the problem and define it as *stu_output*

(3) use the *eval* command to create a file which contains the output of *stu_output*

(4) define the newly created file as *studentfile* and define the file containing the instructor's solution for the problem as *teacherfile* (for clarity and easier reading)

```
for problem in $(seq $number_of_problems)
    do
        sfile=$student/homework$hw_number.txt
        stu_output=$(cat $sfile | grep -E "." | head -$problem | tail -1)
        eval $stu_output > $student/hw_solution_$problem

        teacherfile=instructor/hw_solution_$problem
        studentfile=$student/hw_solution_$problem
```

(5) Use the *diff* command to define a *difference* variable as the number of lines that must be changed in the student's output in order to make it match the instructor's output

(6) Use an *if* statement to append the student's *incorrect*

(6a) If *difference* is greater than or equal to 1 then *incorrect* increases by 1

(6b) If *difference* is greater than or equal to 1 and less than or equal to 3 then *pc* (partial credit) is declared to be 0.5 and is subtracted from *incorrect*

(6c) If *difference* is equal to zero than *incorrect* does not increase

(7) End the Problem Loop

```
        difference=$(diff $teacherfile $studentfile |awk 'BEGIN{count1=0;count2=0}
        {if($0 ~ /^</) count1+=1;
        if($0 ~ /^>/) count2+=1}
        END{if(count1==count2) print count1;
        if(count1 > count2) print count1;
        if(count1 < count2) print count2}')

        if [ $difference -ge 1 ]
            then incorrect=$(echo $incorrect + 1|bc -l)
            if [ $difference -le 3 ]
                then echo Check $student problem $problem for partial credits
                pc=0.5
                incorrect=$(echo $incorrect - $pc|bc -l)
            fi
        fi
    done
```

Return to the Student Loop

Finalize the student's final grade

(1) Define *tot_grade* (the student's total grade) to be the *incorrect* subtracted from the number of problems in the homework

(2) Append the student's *tot_grade* to the homework score sheet

(3) End the Student Loop

```
    tot_grade=$(echo $number_of_problems - $incorrect|bc -l)

    echo The homework grade for $student is $tot_grade out of $number_of_problems

    echo Adding score to HW_Scores_$hw_number
    echo $tot_grade >> HW_Scores_$hw_number
done
```

Appending the Master Table

Use the *paste* command to combine the instructor's master table with the homework score sheet

```
echo Adding homework grades to mastertable
paste mastertable HW_Scores_$hw_number > mastertable.new
mv mastertable.new mastertable
```

Expanding the code

Our reservations in code come mainly from lack of homeworks to test. If we expanded the functionality we would find a more efficient method than creating output files for every problem or at least redirect them to a directory for each homework.

Also we could develop the use of partial credit to be more precise, such as a % of correct lines or perhaps even look at the code itself.

FINAL CODE

```
[adickieson@mis01 ~/groupproject] vi finalproject.sh

#!/bin/bash


echo Enter the homework number
read hw_number

number_of_problems=$(cat instructor/fakesolution$hw_number.txt | grep -E "." | wc -l)

echo "HW_$hw_number" > HW_Scores_$hw_number

for problem in $(seq $number_of_problems)
do
	output=$(cat instructor/fakesolution$hw_number.txt | grep -E "." | head -$problem | tail -1)
	eval $output > instructor/hw_solution_$problem
done

for student in $(cat fakeroster.txt)
do
	incorrect=0
	for problem in $(seq $number_of_problems)
	do
		sfile=$student/homework$hw_number.txt
		stu_output=$(cat $sfile | grep -E "." | head -$problem | tail -1)
		eval $stu_output > $student/hw_solution_$problem

		teacherfile=instructor/hw_solution_$problem
		studentfile=$student/hw_solution_$problem

		difference=$(diff $teacherfile $studentfile |awk 'BEGIN{count1=0;count2=0}
		{if($0 ~ /^</) count1+=1;
		if($0 ~ /^>/) count2+=1}
		END{if(count1==count2) print count1;
		if(count1 > count2) print count1;
		if(count1 < count2) print count2}')

		if [ $difference -ge 1 ]
			then incorrect=$(echo $incorrect + 1|bc -l)
			if [ $difference -le 3 ]
				then echo Check $student problem $problem for partial credits
				pc=0.5
				incorrect=$(echo $incorrect - $pc|bc -l)
			fi
		fi
	done
	tot_grade=$(echo $number_of_problems - $incorrect|bc -l)

	echo The homework grade for $student is $tot_grade out of $number_of_problems
```

```
    echo Adding score to HW_Scores_$hw_number
    echo $tot_grade >> HW_Scores_$hw_number
done

echo Adding homework grades to mastertable
paste mastertable HW_Scores_$hw_number > mastertable.new
mv mastertable.new mastertable
```