Eric Yang

304263623

1. The initial problem I had with this project was finding a way to make the outer loop in the manageOneRound function to work. I had to check if the trial word was between 4 to 6 letters and if the characters were all lower case letters and restart the loop if these conditions weren’t met. I didn’t know how to prevent the loop from moving on to the part that checks if any letters matched the hidden word. I eventually came to the solution of using a counter “restartLoop” that was initialized to 0 at the beginning of each loop. If the trial input was invalid, restartLoop would become 1. All the next steps would only execute if restartLoop was still 0. This let the loop restart when the input was invalid.

The hardest part for me was checking for matching letters in the trial word and hidden word. I found that sometimes the letters would get repeated and counted twice. For example, when I compared eager and eerie, the E’s would get counted 3 times and the program would output 3 instead of the right output, 2. I found the easiest way to handle the problem was to replace the letter in the hidden word with something like ‘0’ whenever a match was found. This way, letters can’t be repeated. I just had to restore the hidden word at the end of the loop by copying the string again.

1. load words into word list array

check if words were loaded

ask user for number of rounds to play

check if rounds is negative

repeatedly for number of rounds:

pick word at random from word list array

repeatedly:

read trial word

check if valid

if trial word matches hidden word

break

else check for matching letters and output number

output round score

calculate and output average, minimum, and maximum scores