1. The project as a whole was not nearly as difficult as project 3 in terms of understanding the spec and organizing the thought processes and literal program most effectively. However, I had a surprisingly difficult time structuring the function I wrote to return the number of planets in a probe word. I ended up writing two separate functions to return the stars and planets. At first, I almost thought I could somehow document every match between the probe and secret word and then call my stars function and subtract it to get the number of planets, but I discovered it wouldn’t necessarily be that simple because there would often be repeats in letters not only in the probe word but also in the secret word. This meant I would double, triple, or even sometimes quadruple count various matches. In the end, I decided I would create two arrays, one for the probe word and one for the secret word. The arrays would hold 100 and 7 zeros respectively, and if there was ever a character used in a match, the positions of those characters would be marked “used” by putting a 1 in the array at that position. That way, it couldn’t be used again resulting in an improper planets output.
2. Program description

Pseudocode

Main

…

Load words

Determine desired rounds

Run that many executions of runOneRound Function topped by round header and word length declaration

Output statistics after each round

…

probeHasCorrectSyntax

…

If the length is too big or small, false

Go through each letter

If anything isn’t a lowercase letter, false

If all checks pass, true

…

Stars

…

Go through each character until the end of the shorter word

If anything matches, add 1 to stars

Return the number of matches

…

Planets

…

Go through each character of the secret word

When it’s not a star,

Go through every character of the probe word to check for matches

If there’s a match, make sure that character isn’t a star

Also, make sure neither have already been used in a match

If so, mark as used and keep looking. If not, add to planets, mark as used, and move to next character of the secret word

When it is a star,

Mark as used and move on

Return the number of planets found

…

probeWordInArray

…

Go through each word in the loaded list

If anything matches, true

If nothing ever matches, false

…

runOneRound

…

Bad argument check causes error

Until guessed correctly,

Ask for probe word

If it has bad syntax or is unrecognized, error (Don’t count against guesses)

If guessed correctly, count as a try and break out

If wrong, count as try, output stars and planets, restart loop

Return the number of tries

…