1. Notable Obstacles

There were two main obstacles that I ran into when doing this project: the use of C strings—especially since I’m more comfortable with C++ strings—and general organization of the code. In order to tackle the first issue, I wrote bits of the code in C++ first and then converted the parts necessary so that the code would work smoothly with C strings. As for the second problem, I decided to split the code into separate parts with “filler code” that I would work on later. For example, I worked on the main routine first and ensured it worked before I created the playOneRound function. When working on the playOneRound function, I left an aspect untouched because I didn’t know how to approach it. Later on, however, I realized that the use of another boolean function would help greatly.

1. Pseudocode

*bool checkWord()*

*check if trial word is correct length*

*check if trial word is all lowercase letters*

*check if trial word is part of word list*

*if not, output response and return false*

*playOneRound()*

*check to make sure parameters are appropriate*

*repeatedly until trial word and mystery word match:*

*request trial word*

*if trial word and mystery word match*

*increment number of tries*

*break*

*if word passes checkWord function*

*increment number of tries*

*create two boolean arrays to keep track of positions for flowers*

*and bees*

*if letters at one position match in the same position*

*increment flowers and alter arrays*

*if letters at one position match with one in another position (and is not included in a bee or flower already)*

*increment bees and alter bee boolean array*

*output number of flowers and bees*

*return number of trial words*

*main()*

*produce array of possible mystery words*

*make sure the array has words, if not*

*output console response and end program*

*prompts user and asks how many rounds to play*

*if round is not positive, output console response and end program*

*repeat for each round:*

*output round header and text telling how many letters are in mystery word*

*call the playOneRound function*

*output console response for number of tries*

*reestablish minimum and maximum attempt values if needed*

*add attempts in this round to total attempts*

*compute and output average (to two decimal points), minimum tries, and*

*maximum tries*