1. Obstacles

The hardest part for me was completely understanding the assignment because of the long project specification. In order to really understand what I needed to code, I grabbed a piece of paper and wrote down an outline of what I needed to create, and how I would go about programming the project.

Furthermore, I had to come up with a way to keep track of which letters had already been used to match for flowers and bees. After thinking for a while, I decided to make two Boolean arrays and mark if a letter had been used by matching the positions and marking the used positions as true.

1. Description of the Design of the Program

Main routine:

I begin the main routine by declaring the 2D char array to store all the loaded words in, and I load the words with the provided getWords() function. If the number returned is not positive, the program says that no words were loaded and exits the program.

I then prompt the user for a number of rounds to play and store it in rounds. Then I use cin.ignore because I want to read strings in next.

* Initialize 3 ints: totalScore to keep track of total score for average, min (for minimum score), and max (for max score)
* if rounds is positive, enter a for loop that runs the same number of times as rounds is equal to.
  + Prints current round
  + Gets a random position with randInt() and prints the length of the word for the user
  + Executes playOneRound which plays a round using the random word at the random position, and returns the score, which is stored in score
  + If the score is negative, immediately go to the next for loop. There was something wrong with the input
  + Otherwise, the score is added to the totalScore
  + Minimum score and maximum score are updated if they need to be
  + Computes the average score
  + Prints how many tries the user took to guess the word, the average tries, min tries, and max tries
* Else: the program tells the user that the number of rounds must be positive, and exits

playOneRound():

* If the inputs are bad, return -1
* Initialize the integer tries, which is a counter for the number of tries a user takes to guess the word
* Enters a for loop that is infinite until the user can guess the word (for(;;))
  + Increments tries and gets a trial word from the user
  + Checks if the inputted trial word is valid by calling my helper method checkWord
    - If checkWord returns 0, that means the word is either not 4 to 6 letters or contains bad characters (non alphabetic or upper case)
    - If checkWord returns 2, that means the word isn’t in the dictionary
  + Then, the program checks to see if the trial word and the mystery word are the same. If they are, it breaks out of the for loop and the program returns tries (the # of guesses)
  + If it gets past that, the function has to check for flowers and bees
  + Initialize two integers: lenMystery and lenTrial which are the lengths of the mystery word and trial word respectively
  + Declare two Boolean arrays of length 101 for trial and mystery because we can assume the length of the inputted trial word won’t be over 100
    - The indexes up to the length of the mystery word for flagM (flag Mystery) are set to false, and the same for flagT for the length of the trial word
  + Declare two integers, flowers and bees, and set them to 0. They are our counters for the number of flowers and number of bees
  + I then make an integer called shorter which stores the shorter value between lenMystery and lenTrial so I don’t get an index out of bounds error when comparing the two strings
  + Flowers takes precedent over bees, so I loop through the length of shorter and compare respective indexes
    - If the indexes are the same, it sets that index in both flagM and flagT to true to mark the letters as taken, and increments flowers
  + Next, we need to find bees, so I have a nested for loop comparing every letter in the mystery word to every letter in the trial word. If the letters are the same and a the Boolean at the respective indexes in the Boolean arrays are false meaning the letters aren’t taken, it sets those spots in the arrays to true and increments bees
  + Then, it outputs the number of flowers and number of bees for the trial word
* This will continue until the user guesses the word correctly

int checkWord(const char words[][7], int nWords, char word[]):

* Inputs: the word list, the number of words in the wordlist, and the inputted trial word
* Defines wordlen to be the length of the trial word
* If the length is less than 4 or greater than 6, the function returns 0
* The function also returns 0 if the word contains uppercase or non-alphabetic characters
* The program returns 1 if the function finds the trial word in the wordlist
* It returns 2 at the end, meaning that the word is not in the word list
* playOneRound uses the values returned by the function to provide error messages or continue the program as needed