**a)** The only real challenge I faced was programming the flowers and bees checker. Initially, I wanted to implement both together inside a nested-for-loop. However, I realized that since the flowers takes priority, I needed to implement it first. My final solution involved a for loop that checked for flowers by comparing the same position for two words up to the length of the smaller word. Then, a nested for loop would check for bees by incrementing the second word while keeping the first one constant to scan for matches. Combined with two temporary arrays that marked whether an element was checked or not already, my fix to the obstacle worked.

**b)** My project works by first creating a character array for the wordlist and filling it using the getWords function. For the number of rounds specified by the user, the program first writes out the round # and mystery word length before running playOneRound. Inside playOneRound, it first checks for string length between 4 and 6 lower case characters before checking if the word is in the wordlist. Then in the flowers and bees checker, two temporary arrays are created as markers. If a flower or bee is found, then the positions in those temporary arrays will be changed from 0 to 1 to mark the position as used, thus preventing duplicate bees or flowers. Finally, the function returns the score and the main function prints out statistics before moving on to another round if specified.

main

create character array for wordlist  
create an int of number of words in wordlist using getWords()   
getWords() fills wordlist with words specified in the file  
  
if nWords is less than 1, write no words loaded and return -1  
  
write how many rounds question  
get input of rounds and clear buffer  
  
repeat for # of rounds

create a random int using randInt()  
write Round #  
write mystery word length  
create an int score equal to playOneRound function  
playOneRound here uses wordlist, nWords, and the random int as parameters  
  
write you got it in “score” try/tries depending on if the score is 1 or not  
  
if round is 1, set min, max, and total to the score  
if else, compare for min, for max, and add score to total  
  
fix to two decimal places  
print out average (total divided by round), minimum, and maximum

playOneRound function implementation

if wordnum from random int is >= nWords, return -1  
 if wordnum is negative, return -1  
 if nWords is not positive, return -1

forever loop  
 initialize temp arrays for trial and original to 0 for all 6 elements  
 create a checker, flower, and bee variable

write trial word  
get input for trial word

check if length of trial word is between 4 and 6  
if not, set checker to 1 and write not 4 to 6 lower case letters

if checker is not 1, check if each character is a lower-case alpha char  
if not, set checker to 1 and write no 4 to 6 lower case letters

if checker is not 1, check if trial word is in the list of words  
if not, set checker to 1 and write don’t know word  
  
if checker was set to 1, return back to asking for trial word

increase score by one  
if trial word is same as original, return score

repeat until the end of shortest length between original and trial

check if the positions in original and trial have the same character  
if yes, increase flowers by one and set both positions in the temp arrays to 1  
  
increment position by one at end of each iteration

repeat until the end of original word, increase position of original each time

if position of temp original is 1, increment position and return to start of loop

repeat until the end of trial word, increase position of trial each time

if characters at the position of the original word and position of the trial word… are the same, AND position of temp word is 0

set temp original at that position to 1  
set temp trial at that position to 1  
increase bees by one  
break out of the inner loop

write flowers # and bees #