1. **Notable Obstacles**
2. The first major obstacle I was faced with was how to count the correct amount of flowers and bees. I decided to create two separate functions for this purpose but this did not solve the fact that the functions I had created kept using characters that had been used on prior flowers or bees. To combat this, I decided to implement two boolean arrays, count and count1, both of length 6, to correspond to both arrays, and made them true when the character in the corresponding element was used.
3. This solution led to my second major obstacle. In using this array within the funcitons, I was leaving the arrays unchanged with each loop of the forloop I implemented in my playOneRound function. This meant that trial words after the first would not display the correct moaunt of bees and flowers because the count arrays were being stuck as true becuase of the prior word. TO combat this I simply initialized all elements in both arrays to false within each loop.
4. Then I had an issue with displaying the correct decimal places for the average, minimums, and maximums. I thought I would try to change the integer storer for points to a double but the effect of this was not what I intended, and division by the amount of of rounds(because my forloop starts at 0) would give undefined behavior. So I moved the points and current round integers into corresponding doubles, and added one to the round in the division to get rid of undefined behavior. This fixed all the issues I was having with it.
5. **Pseudocode**

Int countFlowers(wordGuessed, wordlist, wordIndicator, counted, counted1)

{

Initialize int count to 0;

Until int L reaches length of the goal word:

{

If L is length of the word guessed:

Return value of count;

If element L of guessed word and goal word are the same:

Increase count by one;

Set corresponding element in count and count1 to true;

}

Send count to PlayOneRound;

}

Int countBees(wordGuessed, wordlist, wordIndicator, counted, counted1)

{

Intialize int count to 0;

Until int L which is 0 and increases each loop reaches length of guessed word:

Until int k reaches length of goal word:

If element L in guess is same as element k in goal and L is not k:

Increment count by one;

Corresponding elements in count & count1 are set to true;

Send count to playOne Round;

}

Int playOneRound(wordList, nWords, wordIndicator)**// diff variable names used for clarity**

{

Initialize count and count 1 arrays to false;

If nwords is not positive or wordnum is negative, return -1;

Initialize int tries to 0;

Until int p, initialized as 0 and decreasing reaches 1(never):

Boolean isMath initialized as false;

C string wordGuess initialized;

Output: “Trial Word:”

Input: wordGuess;

If length of guess is not valid or not all character in guess are lower case letters:

Output: “Must be 4 to 6 lower case letters”

Go to next iteration of forloop;

Check if word guessed is in wordList;

If not in wordList:

Output: “unknown word”;

Go to next iteration of forloop;

Increment tries by 1;

If guess is correct:

Send tries to main;

Count arrays initialized to false;

Output: “Flowers: “ and result of CountFlowers “, Bees:” and result of CountBees;

Send tries to main;

}

Int main ()

{

Initialize WordList c string array and FileNameLocation, int MAXWORDS;

Initialize nWords to result of getWords(wordList, MAXWORDS, WORDFILENAME);

Initialize integers points = 0, minPoints, maxPoints;

If nWords is not positive:

output : “No words loaded” and exit program;

Ask for amount of rounds;

If amount of rounds not positive:

Output: “rounds must be positive” and exit program;

Until int i reaches number of rounds inputted by user:

Initialize currentpoints to zero:

Get random number indicator for randomword;

Currentpoints is equal to result of playOneRound;

If currentpoints is 1:

Output: “Got it in 1 try”

Otherwise:

Output: “You got it in” currentpoints “tries.”

If first round:

Min and max set to currentpoints;

Else:

Compare currentpoints to min and max;

Display current average, minimum and maximum;

}