

Report for Project 5: Mystery Word

a. While writing the code for Project 5, I encountered many obstacles but the two most notable obstacles I overcame involved fixing an error in my implementation of the "loadWords" function and writing the code that correctly counted the number of letters in the trial word that are in the mystery word.

The first noteworthy obstacle I overcame was an error message that occurred early on in my work. The error message said "Debug assertion failed!" and indicated the line where the main routine called the "loadWords" function. For the longest time I could not find an error in the function's code. After all, the function's code was given in the project's spec. As I checked to make sure the code of my function was identical to the implementation given in the spec, I remembered two minor changes I had made to the code when copied it from the project spec. The first change was to remove two lines of code that "fixed an issue on Macs" because I have a PC. There was no way that this was the problem. The second change was that I called a different file from a different location with my "loadWords" function. Yes, I had read the notes in the spec that hinted to use a forward slash. After much deliberation I realized the issue. The file I was attempting to call was in .docx format, when it needed to be in .txt format. What a waste of time.

The second notable problem I overcame involved counting the number of letters in the trial that are in the mystery word. My count was sometimes off and sometimes accurate, and it took me a lot of testing to discover why. I realized that when either word had more than one of the same letter that the other word contained as well, my code counted the multiple letters in one word as letters in common even though they were not. I decided to use variable assignment to prevent letters from being counted twice. When my code found a letter that was contained by both words, it executed the statement "trialWord[i] = 1;", which insured the letter would not be counted more than once.

b. Pseudocode:

```
declare libraries and namespace
```

```
declare global constants
```

```
declare external functions
```

```
declare array of C strings
```

```
load array with words
```

```
declare variables in main routine
```

```
give std::rand a different seed value using time function
```

```
if no words loaded in array
```

```
    write no words loaded and end program
```

```
ask how many rounds
```

```
    input number of rounds
```

```
if number of rounds is not positive
```

```
    write that it must be positive
```

```
repeatedly for the number of rounds inputted:
    assign a random integer to an int
    write out the round number
    write out the length of the mystery word
    execute a round of the game
    write out the number of tries it took to win
    assign statistics to variables and write them out
```

```
function that loads words into array (do not understand implementation)
```

```
function that executes one round of game
    if number of words is not positive, word number is less than zero, or
    word number is greater than or equal to number of words
        return -1
    declare C string and initialize variables
    repeatedly while trial word is not equal to mystery word:
        ask for trial word
        input trial word
        increase attempt counter by one
        if trial word is equal to mystery word
            return number attempts
        if trial word has less than 4 or more than 6 letters or is not
        all lowercase letters
            write out error message
        if trial word is between 4 and 6 letters and is all lowercase
        letters
            check to see if word is in array
            if no
                write out error message
            if yes
                count and write out number of letters in common
```

```
function that produces a sequence of numbers that are almost independent of
eachother
```

```
function that goes through each character of C string to see if they are
lowercase letters
    return true if they are all lowercase letters
    return false if they are not
```