Or Brener CIS* 3190 Software for Legacy Systems Date: February 6th 2023
Student ID#: 1140102 Course Instructor: Michael Wirth

Assignment 1
Word Jumble Solver in Fortran
Reflection Report

Program design:

1. The first step is to build a dictionary using the buildLexicon() subroutine
   - The dictionary is a dynamically allocated array of strings
   - The dictionary is built from the given text file "dict2.txt"
   - I also store all the line numbers (indices) for each letter in the alphabet (will be used to speed up searching)
2. Then I prompt the user to enter the number of jumbled words they would like to enter. Then I do the follow n number of times; where n is the number of jumbled words to be inputted
3. I generate all the anagrams (permutations) of the given jumbled word
   - This uses the generateAnagram() recursive subroutine
   - Storing the anagrams in a dynamically allocated array of strings
   - Number of anagrams is calculated by the factorial of the length of the original string
4. For all the anagrams, I search them in the dictionary using the findAnagram() subroutine
   - I traverse the dictionary array at the index of the first letter of the anagram to the next letter (e.x anagram "hloel" will search the dictionary for all words starting with "h")
   - If a match is found it stores the match in the string foundAnagram
5. For each found anagram (taken from the users' inputted jumbled word), I inquire how many circled letters are in that word, and get the indices of said circled letters
6. I concatenate all the circled letters into one final jumbled word
7. Finally, I solve the final jumbled word by generating all of it's anagrams and finding a match in the dictionary
   - Same as steps 3 and 4

Notes:

- If a jumbled word cannot be found in the dictionary, it prints that no match was found
- Note that for all instances where user input is prompted, accurate input validation is executed (i.e input type, logical values…)

Programming Process/Reflection:

Fortran is definitely a headache to study and program in (it is a legacy language after all). The hardest challenge that I found was that it was quite difficult to find helpful resources online when running into bugs (although the professor's website was definitely a great resource!). With that being said, since Fortran is a precursor to C, and I have 3+ years of experience of writing in C, and coupled with the fact that the general paradigms of the language are similar with the few differences outlined in the course text, writing in Fortran did not feel like I was learning a new language from scratch.