In this lab, you will develop an application to solve word jumble puzzles. A word jumble is a common type of puzzle where the letters of a word have been shuffled around, and it is up to the player to decode the original word.

When your program begins, it should open the file "dictionary.txt" and read all of the words in this file into a vector. (This file contains ~4000 common English words.) Then, it will wait for the user to type in a word and scan the vector of words to find any words that match. If it finds one or more words with the same letters that the user typed in, it should print \*all\* matching words, one per line. If there is no word in the dictionary that has the same letters as the word the user typed in, the program should print the message "No matches" on a line. Either way, the program should continue to prompt the user for words and print all matches until the user types the word "quit". Matches for "quit" should not be printed before the program exits.

In order to detect whether two words are a jumble of one another (a "match"), I recommend creating two arrays of size 26 and counting how many a's, b's, c's, etc., that each word contains. If both words have the same number of all 26 letters, they are jumbles of one another. Your code will not be tested on input that has uppercase letters, special symbols, punctuation, or digits; you may assume that the user will only type lowercase letters. That being said, you may wish to add error checking for your own purposes.

I've written some baseline code that you can use. This code defines several functions that you may find helpful to implement when solving this lab, described below. You are \*not\* required to implement these specific functions. You may solve the problem however you like; however, your program should match the expected output exactly--it should only print matched words (or the message "No matches") when it runs. The words it prints should be solutions to the jumble, of course.

* void countLetters(string word, int\* arr): this function counts how many a's, b's, c's, etc., are in the given string, and writes the 26 letter counts into arr (arr[0] = # a's, arr[1] = # b's, etc.)
* bool isJumble(string word1, string word2): this function returns whether or not the two words are jumbles of one another
* vector<string> findMatches(string word, vector<string> vec): this function returns a vector containing all of the words in vec that are jumbles of the given word

You should submit a single source (.cpp) file containing your solution to this assignment. Your code must be able to compile using g++. I strongly recommend that you install VS Code (and MinGW, for Windows users). Submissions that do not compile will receive a substantial penalty. This assignment is due Thursday, June 4.

*Hint:*  you can subtract 'a' from a lowercase letter to calculate which position in the word count array corresponds to that letter.