Lab 14: End of File (EOF)-controlled loops **Due:** 10/31/24

In this lab you will practice using an End-Of-File while loop to perform repetitive tasks.

EOF-controlled while loops are a special case of flag-controlled while loops. When the end of a file is reached, the end of file flag is set to true. The example program demonstrates how to test the end of file flag.

Example Program: EOF-Controlled while loops

The example program for EOF-controlled while loops reads a series of numbers from an input file and finds the average of those numbers. The **eof-while-numbers.cpp** program provided illustrates the skills you are learning in this lab. Open it in your IDE, read and understand the code and finally run the program with different inputs to see how it works.

Your Program

For your program you will search a dictionary file for the word provided by the user. First ask the user to input a word. Search the dictionary file for that word and tell the user whether the word was found.

This program demonstrates the linear search technique. When you perform a linear search, you start at the beginning of a list and search through the list in order until you reach the desired item or the end of the list (**Hint** - use break to stop the program when you find the desired item). Linear search is very inefficient, and we will learn about better search techniques later in the semester.

Open **lab14.cpp** in your IDE and implement the algorithm provided in the source code as comments. Use the provided dictionary file **dictionary.txt** to test your program.

Sample runs of my program

```
Enter a word to search in dictionary: adds

Word adds found in line 23
```

```
Enter a word to search in dictionary: ada

Word ada is not in the dictionary
```

I am posting my sample runs for your reference. Please run your program as many times as necessary to ensure that you tested it thoroughly. Make sure it behaves like my sample runs.

Don't forget to include at the top of the programs, the comments shown below with your information (name, class and section number, etc.)

```
////////////////////////////////////////////////////////////////
//
// Name: <Put your name here>
// Date: <Today's date>
```

```
// Class: <Your class number and section number, like: CSCI 1470.02>
// Semester: <This semester, like: Fall 2012>
// CSCI 1470 Instructor: <Your lecture instructor's name>
//
// Program Description: Enter here your description of what the program does
//
////////////////////////////////////////////////////////////////////
```

When done, submit your solution through Blackboard using the "Assignments" tool. Do Not email it.

Paste the link to your final solution along with your source code in the textbox opened when you click on Create Submission before you click on Submit.

The following is the basic criteria to be used to grade your submission:

You start with 100 points (each is worth 50 points) and then lose points as you don't do something that is required.

-5: Minor mistakes
Missing comments at the top of the program
-10: Moderate mistakes
Doesn't properly open input file (for example, doesn't check if it was opened)
Doesn't properly close input file
Doesn't properly handle case when word is found
Doesn't properly handle case when word is not in dictionary
-20: Major mistakes
Doesn't stop getting input when end of file is reached
program does not implement the provided algorithm
Incorrect/missing source code
or incorrect/missing link to your solution
-30: Program doesn't run to completion
-40: Program doesn't compile
-100: The code submitted is not your creation (you got it from a web site or another person)
-10: Late

Note: more points may be lost for reasons not specified here.