CS250 Assignment 2
Word Search

Date assigned: Friday, February 8, 2013
Date due: Part I: Wednesday, February 13, 2013
Part II: Wednesday, February 20, 2013
Total points: 40

Problem

You are to write a complete C++ program that will find "hidden" words in a rectangular array of letters.

Your program will first read the number of rows and the number of columns in the puzzle followed by the puzzle itself (the rectangular array of letters). Next, your program will read the list of words to be searched for in the puzzle into an array of structs.

As output, your program must first echo print all of the data read by the program properly labeled. Next print each of the words and whether the word was found or not. For each word found include the following information in table form:

1) word
2) row found
3) column found
4) word's orientation (horizontal, vertical, diagonalLR, diagonalRL)

If the word was not found, print:
1) word
2) the message NOT FOUND

In writing your program you may assume that each word occurs at most once (but note that a letter in the puzzle may be part of more than one word). You may assume that the number of rows and columns will not exceed 20 each and the number of words to be searched for will not exceed 25. Also, each of the words to be searched for will not contain more than 32 characters (including the null character) and will be left justified in the file (one per line).

Your program must search for words in the following ways:

1) horizontally (left to right)
2) vertically (top to bottom)
3) diagonally (upper-left to lower-right)
4) diagonally (upper-right to lower-left)

Consider the following puzzle.txt data file:

```
6 4
THAY
EHRJ
FZEO
YHUB
IJGH
NHIB
THAT
THE
HIP
JOB
```

Correct output results must look exactly like the following (heading and all):

```
******************************
*        Find A Word         *
******************************

Number of Rows:     6
Number of Columns:  4

Puzzle
------
THAY
EHRJ
FZEO
YHUB
IJGH
NHIB

Words
-----
THAT                            NOT FOUND
THE                             FOUND (Row: 1 Column: 1 Diagonally UL to LR)
HIP                             NOT FOUND
JOB                             FOUND (Row: 2 Column: 4 Vertically)
```

Notes

1. Minimally, a data file must consist of at least the number of rows and columns of the puzzle.
2. I will run your program on the test data supplied above and few more files, so make sure you test all of the border cases.
3. The input file is called puzzle.txt.
4. Output your results to the display screen.
5. Do not use the data type string from the library #include <string> anywhere in your program.

To complete this assignment you must

1. **USE FUNCTIONS IN A MEANINGFUL WAY.** Remember, a function is to have a single well-defined purpose. Example, reading in a puzzle. In solving this problem, there is an OK solution, a pretty good solution, and "THE" solution. Please try and find "THE" solution.
2. Create a new C++ project in Visual Studio 2010 called **02_WordSearch**. The new project is to be added to your existing solution **PUNetIDAssignments.**
3. Type the solution (**fully documented/commented**) to the problem into your project.
4. Remember to enter in your name as the author of the program.
5. Make sure that your program compiles and runs correctly. If you get any errors, double check that you typed everything correctly. Be aware that C++ is case-sensitive. Also, there must not be any warnings when compiling your program or you will lose points.
6. Once you are sure that the program works correctly, it is time to submit your solution. You do this by logging on to Turing and placing your complete solution folder with two working projects (01_DNA and 02_WordSearch) in the proper **CS250 Drop** folder. Make sure that you copy your program folder and don't move the folder. If you move the folder, then you will not have your own copy!

Additional Notes

1. You must follow the coding standards found on the main CS250 Web page.

2. You must use constants when possible.

3. Your program will be graded on **efficiency**. In other words, you will be marked down for repeating code statements unnecessarily.

4. You may only use the C++ programming concepts covered thus far in class. Do not use any more advanced concepts that we have not covered or any other programming concepts that you have had experience with.

5. Your output must look **exactly** like the sample given.

6. If this program sounds difficult, it's not that bad if you get an EARLY start. Make sure you understand all of the pieces before beginning to code your solution. Code your solution one function at a time not all at once as doing so makes for much smoother debugging.

Remember, this is an individual assignment. Refer to the syllabus for assignment policies

By February 13, you are to have coded up and turned in a working program that has at least four functions completely working as follows:

1) readPuzzle - reads the puzzle from the datafile into the 2D array.

2) printPuzzle - prints the puzzle to the display screen

3) readWords - reads in the words to be searched for into the array of structs

4) printWords - prints the words and whether they are found or not.

Note: Since this part of the assignment is not doing any searching, all words will be NOT FOUND, so the results of running this program on the above data file will produce the following results:

```
******************************
*        Find A Word         *
******************************

Number of Rows:     6
Number of Columns:  4

Puzzle
------

THAY
EHRJ
FZEO
YHUB
IJGH
NHIB
Words
-----

THAT                           NOT FOUND
THE                            NOT FOUND
HIP                            NOT FOUND
JOB                            NOT FOUND
```

Once you have finished and tested this part of the assignment, turn in your project and a colored hardcopy by the beginning of class on February 13 and then write the rest of the functions to complete the remainder of the assignment. I do not want a partially completed Part II turned in.

Remember, your completed program is due on Monday, February 20 by the beginning of class.