Word Search (Due 12 Apr 2013)

Given an *m* by *n* grid of letters, and a list of words, find the location in the grid where the word can be found. A word matches a straight, contiguous line of letters in the grid. The match could either be done horizontally (left or right) or vertically (up or down). We will not consider words that can be matched diagonally.

The input will be in a file called *hidden.txt* and your output will be written to a file called *found.txt*. The format of the input file will be as follows:

- First line will have two integers - m, the number of lines in the grid and n, the number of characters in each line. The integers *m* and *n* will be separated by one or more spaces.
- There will be a single blank line.
- There will be *m* lines, where each line will have *n* characters, all in upper case, separated by a space.
- There will be a single blank line.
- There will be a single integer *k*, denoting the number of words that follow.
- There will be *k* lines. Each line will contain a single word in all uppercase.

There will be *k* lines in your output file. Each line will have the word that you were search for followed by two integers *i* and *j* separated by one or more spaces. The numbers will be right justified and aligned. The number *i* gives the row and the number *j* the column of the **first** letter of the word that you were required to find. Rows and columns are numbered conventionally, i.e. the first row is 1 and the first column is 1. If you do not find a word in the grid then the values for *i* and *j* will be *0* and *0*.

Extra Credit (10 points): Find all the diagonal words. The diagonal words could go either right to left or left to right along any diagonal. Give the row number and the column number of the first letter of the word.

The file that you will be turning in will be called **WordSearch.java**. The file will have a header of the following form:

```
/*
  File: WordSearch.java

  Description:

  Student Name:

  Student UT EID:

  Course Name: CS 312

  Unique Number:

  Date Created:

  Date Last Modified:

*/
```

You will follow the standard Java Coding Conventions. You can either view the HTML page or download the PDF or Postscript and print it out. There is a modification that I would like to make to the standard coding conventions. Please align the opening and closing braces vertically so that you can easily make out the blocks of code. For example:

```
Do this:
if ( x > 5 )
{
  a = b + c;
}

Not this:
if ( x > 5 ) {
  a = b + c;
}
```

Use the turnin program to submit your **WordSearch.java** file. The proctors should receive your work by 11 PM on Friday, 12 April 2013. We will be looking for clean logic and good documentation. There will be substantial penalties if you do not adhere to the guidelines.

- You must submit the .java file and not the .class file.
- Your .java file should have the header with the proper documentation.
- You should be submitting your .java file through the web based *turnin* program. We will not accept files e-mailed to us.
- Your code must compile before submission.