W V E R T I C A L L
R O O A F F L S A B
A C R I L I A T O A
N D O D K O N W D C
D R K E S O O D D K
O E E P Z E G L I W
M S I I H O A E R A
A L R K R R I R E R
K O D I D E D R C D
H E L W S L E U T H
Week
Find
Random
Sleuth
Backward
Vertical
Diagonal
Wikipedia
Horizontal
Word Search

FINDERS KEEPERS

OVERVIEW

Have you ever played a 'word search' game where the objective is to find a bunch of words hidden somewhere in a 2D table of letters? The words can be diagonal, horizontal, or vertical and even be upside-down. In this assignment, you will write a program to play a game of word search.

DETAILS

You must write a WordSearch class that reads a **word-search** file and prints the solution to an output file.

WORD-SEARCH FILE FORMAT

A word-search game consists of two parts: a table of letters that we refer to as the "puzzle" and a list of words that we refer to as the "word-list". A word-search file contains the puzzle followed by the word-list as described below.

- The first line is exactly the phrase "#WS-FILE".
- The second line starts with zero-or-more whitespace characters; followed by an integer W denoting the puzzle-width; followed by one-or-more whitespace characters; followed by an integer H denoting the puzzle-height; followed by zero-or-more whitespace characters
- The next H lines contain exactly W capital letters.
- Each of the following lines starts with zero-or-more whitespace characters; followed by a word in the word-list. This pattern may repeat itself within a single line (i.e. each line has one-or-more "zero-or-more whitespace followed by word" patterns). A word is a sequence of one-or-more letters (either upper or lower case). There is no requirement that words be meaningful (as in 'defined in an English dictionary').

WORDSEARCH

Write a WordSearch class that has a 'main' method that can be executed from the command-line. This method will be given two command-line arguments: the name of a word-search file followed by the name of the output file. For every word in the word-list the program must output whether the word is in the puzzle and, if the word is in the puzzle, where the word can be found. For each word in the word list, your program must print exactly one line. The line that you print must be exactly one of the following phrases. Note that items in angled-brackets denote place-holders that will be replaced by actual values.

- <WORD> NOT-PRESENT
- <WORD> (<X>, <Y>) <DIRECTION>

<WORD> is a word in the word-list but printed in all upper-case. Searching for words in the puzzle is done without regard to case (i.e. it is a case-insensitive serach). The phrase (<X>, <Y>) gives the starting location of the word within the puzzle. <X> is an integer that denotes the column (starting with 1 at the left and increasing to the right) and <Y> is an integer that denotes the row (starting with 1 at the top and increasing downward). <DIRECTION>

denotes the orientation of the word with respect to the starting location. <DIRECTION> is one of the words "RIGHT", "LEFT", "UP", "DOWN", "UP-RIGHT", "UP-LEFT", "DOWN-RIGHT", "DOWN-LEFT". If <WORD> is of length one, then <DIRECTION> must be omitted.

The word-list must be printed in the same order given in the word-search file. If there is more than one occurrence of the word, you must print the first occurrence. The first occurrence is the one that starts in the highest-leftmost location in the puzzle and whose direction occurs first in the list "right", "down-right", "down", "down-left", "left", "up-left", "up", "up-right").

ERROR CONDITIONS

In the event of an error, your program must print a message to the terminal window. Your program must not create an output file of any kind if any error occurs. The errors and messages are described below

1. If there are not exactly two command-line arguments, then print "Usage: <word-search> <output>".
2. If the input file does not exist then print "Error. The word-search file does not exist".
3. If the input file is not a word-search file or an error occurs when reading from the file, then print "Error reading the word-search file".
4. If some error occurs when creating the output file, then print "Error creating the output file".

Note that your program must either 1) create a correct output file OR 2) print the above error message; it must never 'crash' with an unhandled exception of any kind.

EXAMPLE INPUT FILE

```
#WS-FILE
       5                 5
EAGLE
EEBAB
WTSEB
EVOLE
TEATE
BAGLE    BEE     LOVE
vOLe
     Ate A
EAT Tea BesT
 aosb                    SEE
```

EXAMPLE OUTPUT FILE

```
BAGLE NOT-PRESENT
BEE (3,2) DOWN-RIGHT
LOVE (4, 4) LEFT
VOLE (2, 4) RIGHT
ATE (3, 5) RIGHT
A (2, 1)
EAT (2, 5) RIGHT
TEA (2, 3) UP
BEST (5, 3) LEFT
AOSB (3, 5) UP
SEE (3, 3) UP-RIGHT
```