CSI3108-01 2016. 09. 05

Programming HW#1

Max 30 points

Due on Sept. 9 (Fri), 2016, by 5 pm

Write a Java program to search given words from a dictionary of NxN 2-dimensional array of chars as shown below.

	0	1	2	3	4	5	6	7	8
0	i	c	o	n	q	u	e	r	w
1	n	r	o	u	d	e	e	p	a
2	t	r	o	g	r	l	d	s	r
3	e	g	g	g	p	z	y	h	e
4	r	u	j	p	l	i	n	e	a
5	s	y	a	n	p	a	a	a	s
6	d	i	v	i	d	e	m	p	o
7	r	e	a	l	q	i	i	h	f
8	n	o	m	e	l	x	c	i	t

Your program is to find words from the dictionary that can be spelled out in the array by starting at any character, then moving in a straight line up, down, left, right, or on any of the four diagonals, wrapping around the array. For example, the above array contains the word 'algorithm' because it starts at (5, 5) and ends at (6, 6); and it contains the word 'greedy' since it starts from (3, 3) and ends at (5, 1).

Input The first line has two integers N and M, where N the array size and M is the number of words to be searched, N ≤ 50, M ≤ 20. Then N lines of N characters each (with each character separated by a single space), followed by the M words (each word per line) in lexicographic order. Also assume that all characters in your programing are only lowercase. Finally, you may assume that all of the words in the dictionary are at least 4 characters long.

Output format. Print out the starting and ending element indices of the dictionary for a word per line, as shown in the **Sample Output**. If a word appears in the puzzle more than once, print out the one with 'lexicographically first' starting indices; for example, the same word starts at (3, 0) and at (2, 40), select (2, 40) for printing, since 2 is in front of 3 no matter what in the column index is.

If a word to be searched is NOT in the dictionary, simply print '0'.

<u>Sample Input</u>

```
9 10    <---   N = 9, M = 10; 9x9 array of chars and 10 words are to be searched.
i c o n q u e r w
n r o u d e e p a
t r o g r l d s r
e g g g p z y h e
r u j p l i n e a        9x9 dictionary
s y a n p a a a s
d i v i d e m p o
r e a l q i i h f
n o m e l x c i t
algorithm
conquer
divide
dynamic
greedy                   10 search words are given lexicographically.
heap
java
linear
memory
software
```

<u>Sample Output</u>

```
5 5 6 6     <--- 'algorithm' starts at (5, 5) and ends at (6, 6).
0 1 0 7     <--- conquer
6 0 6 5     <--- divide
2 6 8 6     <---dynamic
3 3 5 1     <--- greedy
3 7 6 7     <--- heap
4 2 7 2     <--- java
4 4 4 0     <---linear
```

```
0          <---‘memory’ is NOT in the dictionary.
5 8 3 8    <---software
```