Dr. Tolga Soyata. *© University at Albany, Dept. of ECE,* *tsoyata@albany.edu*

ICSI 213 Homework 2 (9 regular + 2 bonus pts)

Due Date: July 26, 2017 11:59PM

Instructions for completing Homework 2:

- There is a 4 point/day penalty for late submissions. No negotiations! We have to stick to it!
- You do not get the bonus points if you submit late. No negotiations!
- You are absolutely not allowed to look at anybody else's report or drawings or anything. Don't even communicate with them for ideas. Any similarity between any two reports will go against "Academic Honesty" rules. You must do this homework (and all of the other ones) on your own.
- It is possible that there is some example code out on the web. You are absolutely NOT allowed to copy-paste somebody else's code as this would be considered plagiarism. This will result in a grade of zero from HW2 and potentially more serious action, as specified by academic honesty guidelines. Furthermore, you will not learn anything if you do this!

Homework 2 Required Submission

- **MS Word report:**
  - **Lastname-HW2Report.docx**
  - Upload on Blackboard under "**Homework 2 Submission Area**." This report will contain your answers to all of the questions.
  - For example, if your lastname is Smith, your report will be uploaded on Blackboard as Smith-HW2Report.docx.
  - Include plenty of screen shots, drawings, write-ups to make it very clear what you did.
- **.java Files:**
  - Upload your Java code on Blackboard separately from the report.
  - If there are multiple .java files, ZIP them together into one .ZIP file.
  - In your report, describe your naming convention for your Java code files.
- **Commit History**
  - Upload a screenshot of your Git commit history.

UNIVERSITY AT ALBANY
State University of New York

solvePuzzle Program Description:

We have a 20x20 puzzle in which each cell contains a letter. We want to write a program that looks for some keywords in this puzzle and use recursion as the method for solving this problem.

The puzzle is a CSV file ("puzzle.csv") and looks like this:

```
x,t,b,n,t,e,l,u,b,n,s,p,y,z,l,g,f,q,y,d
j,o,k,j,e,c,d,m,n,j,v,l,u,n,p,t,b,j,h,u
i,n,y,v,r,s,f,m,e,b,s,o,w,y,a,z,u,t,b,e
p,e,g,s,f,y,f,n,y,d,n,o,a,x,q,b,s,j,s,k
f,h,l,a,k,r,h,m,n,z,l,a,f,f,u,c,l,b,s,x
l,c,h,j,a,a,p,h,y,r,e,z,v,d,b,r,c,a,w,h
y,b,s,t,b,c,s,v,u,l,u,y,f,v,c,h,a,x,f,i
c,m,y,n,o,u,l,e,t,b,z,g,x,s,j,o,k,j,t,k
y,o,f,c,n,s,r,s,x,l,b,x,m,k,o,m,p,v,x,a
k,n,c,o,i,e,e,i,b,i,y,k,z,c,c,a,k,u,v,t
f,t,o,i,w,l,p,r,q,p,y,n,h,c,u,s,d,k,i,k
l,x,z,x,k,k,i,n,o,o,j,r,e,w,s,a,j,j,k,p
p,y,i,j,w,t,a,f,c,w,c,e,q,y,x,c,u,a,s,p
s,y,j,i,r,g,p,m,h,e,s,t,i,c,f,q,y,c,q,a
w,d,i,o,t,a,m,s,t,h,d,g,a,a,v,q,o,u,p,s
u,v,j,l,s,o,h,k,c,d,u,y,b,z,v,f,k,f,p,p
w,e,o,t,b,i,t,d,u,l,h,d,a,y,l,b,v,z,w,a
e,b,j,c,c,w,d,z,i,v,o,a,k,m,l,j,v,s,n,o
y,v,m,x,d,o,g,l,f,o,l,h,o,v,k,p,z,n,g,f
p,w,o,t,w,p,i,s,v,e,b,a,r,d,e,m,e,w,p,c
```

Note that there are four hidden words in this puzzle: "albany", "syracuse", "rochester", and "buffalo"

x	t	b	n	t	e	l	u	b	n	s	p	y	z	l	g	f	q	y	d
j	o	k	j	e	c	d	m	n	j	v	l	u	n	p	t	b	j	h	u
i	n	y	v	r	s	f	m	e	b	s	o	w	y	a	z	u	t	b	e
p	e	g	s	f	y	f	n	y	d	n	o	a	x	q	b	s	j	s	k
f	h	l	a	k	r	h	m	n	z	l	a	f	f	u	c	l	b	s	x
l	c	h	j	a	a	p	h	y	r	e	z	v	d	b	r	c	a	w	h
y	b	s	t	b	c	s	v	u	l	u	y	f	v	c	h	a	x	f	i
c	m	y	n	o	u	l	e	t	b	z	g	x	s	j	o	k	j	t	k
y	o	f	c	n	s	r	s	x	l	b	x	m	k	o	m	p	v	x	a
k	n	c	o	i	e	e	i	b	i	y	k	z	c	c	a	k	u	v	t
f	t	o	i	w	l	p	r	q	p	y	n	h	c	u	s	d	k	i	k
l	x	z	x	k	k	i	n	o	o	j	r	e	w	s	a	j	j	k	p
p	y	i	j	w	t	a	f	c	w	c	e	q	y	x	c	u	a	s	p
s	y	j	i	r	g	p	m	h	e	s	t	i	c	f	q	y	c	q	a
w	d	i	o	t	a	m	s	t	h	d	g	a	a	v	q	o	u	p	s
u	v	j	l	s	o	h	k	c	d	u	y	b	z	v	f	k	f	p	p
w	e	o	t	b	i	t	d	u	l	h	d	a	y	l	b	v	z	w	a
e	b	j	c	c	w	d	z	i	v	o	a	k	m	l	j	v	s	n	o
y	v	m	x	d	o	g	l	f	o	l	h	o	v	k	p	z	n	g	f
p	w	o	t	w	p	i	s	v	e	b	a	r	d	e	m	e	w	p	c

Part I: Reading the file, and getting a keyword from the user (1 point)

Writing a method that reads the CSV file and receives a keyword from the user to search for, will get you 1 point.

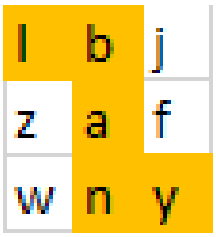
Part II: Searching for the keyword in the puzzle recursively (5 points)

You need to find the given keywords in the puzzle. The search process should follow these steps:

- You need a loop (maybe a nested loop) to find the first character (of each keyword). For example, if you are looking for 'albany', go through the cells one by one to find an 'a'.
- When the first character is found, look for the rest of the word recursively around the position of the first word. For example, assume that when looking for 'albany' you found an 'a' in location (4,3). Then you need to look for 'lbany' in locations (3,3), (3,2), (4,2), (5,2), (5,3), (5,4), (4,4), and (3,4).
    - Note that the first number in the (x, y) pair is the column and the second number is the row.

(1,1) x	(2,1) t	(3,1) b	(4,1) n
(1,2) j	(2,2) o	(3,2) k	(4,2) j
(1,3) i	(2,3) n	(3,3) y	(4,3) v
(1,4) p	(2,4) e	(3,4) g	(4,4) s

- If the second letter is found in the neighborhood of the first one, keep on searching for the rest of the word, if not, go back to the loop and keep searching for the first character.
- Remember that you need to keep track of the cells that are part of the word and not count a cell twice. For example, the following photo does **not** count as 'albany', because it uses the same 'a' for '**a**lbany' and 'alb**a**ny.'



Part III: Highlighting the keyword in the output (3 points)

Create a JFrame output that shows the whole puzzle and highlights the keywords graphically, each keyword must be a different color. Assume that there are a maximum of 4 keywords, requiring only 4 colors (red, green, orange, yellow). When two keywords use the same letter, always make them blue.

BONUS: Making the puzzle (2 points)

Build a method that receives 4 keywords from the user and then uses them to generate a puzzle like the puzzle that is given to you in this assignment. You must make sure that:

- The keywords are positioned in the puzzle randomly, as shown before. They can have any shape, as long as they cross each other and make "knots."
- All of the cells that are not part of the given words are filled with random letters.

Your method should produce a CSV file for this puzzle and solve the puzzle as the last step.