CMPSCI 383, Fall 2011 Homework 3 Due in class or in the main office of the Computer Science building by 4:00 PM, November 10, 2011

Problem 1: (10 points) Exercise 13.8 on page 507Problem 2: (10 points) Exercise 13.15 on page 508Problem 3: (10 points) Exercise 13.17 on page 508. This problem has a typo. When they say *B*, they mean *Y*.

Programming Assignment: (70 points) For this assignment, you will create a program that solves crossword puzzles. Consider the following puzzle:

	1	2	3	4	5	
1	+	+	+ 2	+ 	+ 3	+
2	+	+ # +	 	+ # 	+	
3	# 	4	' L	5		
4	6	' # +	7 	 +	 +	
5	8					
6	+====	# +	# 	 	# +	- +
						E

The # symbols denote walls (no letters should be placed there). The numbers denote the starting points of different words. Words 1,4,7, and 8 are *across* words, and 2,3,5, and 6 are *down* words. Your program should read from the file *in.txt* a list of words, such as the following:

AFT	LASER
ALE	LEE
EEL	LINE
HEEL	SAILS
HIKE	SHEET
HOSES	STEER
KEEL	TIE
KNOT	

The list of words will be in alphabetical order, and will use only capital letters. There is no constraint that a word can only be used once. Your program must figure out which word starts in each of the 8 numbered cells. Dowload the file *www.psthomas.com/Data/in.txt*, which contains the list of words above, and run your program using it. You must read the words from an input file, as we will be using a different input file to test your programs.

You must implement 4 different variants of BACKTRACKING-SEARCH (where MRV denotes the minimum-remaining-values heuristic and LCV denotes the least-constraining value heuristic, see pages 216-217):

- Backtracking search without MRV and without LCV
- Backtracking search with MRV and without LCV
- Backtracking search with MRV and with LCV
- Backtracking search without MRV but with LCV

For each of the above, you must print out the number of calls to the recursive function BACKTRACK. In all four cases, your implementation of BACKTRACKING-SEARCH should use forward checking. Assume that the initial domain of each variable includes only those words of the correct length. When selecting which variable to assign next, ties should be broken by selecting the smallest number first. When selecting which value to assign next, ties should be broken by selecting the word that comes first alphabetically.

In order to test your code, we must be able to run it. So, do *not* submit a hard copy of your code. Instead, place your code on Edlab and make sure that it is working there.

You should submit:

- For the word list above, list what the variables are, and what their domains are.
- List all of the constraints.
- Detailed instructions for how to compile and execute your program.
- A brief description of all of the major functions and classes in your code.
- The number of calls to BACKTRACK for each of the four different cases.

You do not need to submit your program's output on paper. However, when executed, your program should print the numbers 1 through 8 (in order), followed by the word that should be assigned to that number.

Extra Credit: (20 points) Implement AC-3, and use it to prune the domains of each variable before running BACKTRACKING-SEARCH. Rerun all 4 variants of BACKTRACKING-SEARCH, and list the number of calls to BACKTRACK for each (using AC-3 before hand).