PROJECT TWO

REQUIREMENTS:
For this assignment, you are going to solve the famous "zebra puzzle" (also known as the "Life International Puzzle", rumored to have been proposed by Albert Einstein. You will need to formulate the problem as a constraint satisfaction problem (CSP). Solve using any of the methods discussed in class.

Here is the puzzle (note, this is pasted directly from Wikipedia:
http://en.wikipedia.org/wiki/Zebra_Puzzle

1. There are five houses.
2. The Englishman lives in the red house.
3. The Spaniard owns the dog.
4. Coffee is drunk in the green house.
5. The Ukrainian drinks tea.
6. The green house is immediately to the right of the ivory house. (*your* right)
7. The Old Gold smoker owns snails.
8. Kools are smoked in the yellow house.
9. Milk is drunk in the middle house.
10. The Norwegian lives in the first house.
11. The man who smokes Chesterfields lives in the house next to the man with the fox.
12. Kools are smoked in the house next to the house where the horse is kept.
13. The Lucky Strike smoker drinks orange juice.
14. The Japanese smokes Parliaments.
15. The Norwegian lives next to the blue house.

`Now, who drinks water? Who owns the zebra?`

Other notes:
- Each of the five houses is painted a different color; each inhabitant of each house is of a different nationality, owns different pets, drink different beverages and smoke different brands of American cigarettes.
- It is easy to find the solution online to the problem. Therefore, you should be able to easily check to see if your CSP solver works
- To get you started, for example, you could have the following variables:

  X = {house_color, nationality, drink, smoke, pet}

  with the following domains, respectively:

  D = {yellow, blue, red, ivory, green}, {Norwegian, Ukrainian, Englishman, Spaniard, Japanese}, {Water, Tea, Milk, OrangeJuice, Coffee}, {Kools, Chesterfield, OldGold, LuckyStrike, Parliament}, {Fox, Horse, Snails, Dog, Zebra}

- The constraint set, *C*, well… I'll let you do this!

PROGRAM INPUTS AND OUTPUTS:
For this program, it is okay if you represent all of the inputs (the problem formulation, variables, domains and constraints) internally (AKA hard-coded). However, your program should output (to the console) each variable and the remaining values in its domain. Note that this should indicate the solution (i.e. you should not have two values in the domain left for *pet*). I do not care what format you use to output this information, so long as it is clearly readable.

ADDITIONAL REQUIREMENTS:
Your program must be robust. For example, your program should not crash and burn if the input is invalid. Reject any invalid input with an appropriate error message and termination.

There should be some elegance to the code you write and must contain some documentation. Your may lose a point if your code looks careless/confusing and/or sloppy.

PROGRAMMING LANGUAGES ALLOWED:
You may use any standard language (Java, C, C++, or Python) to write this program. If you wish to use something non-standard, please me prior to doing so. You need to be sure that I can compile and run your program.

Super Mega-Huge Insane HINT:
Python has a CSP solver library. If you know Python, it is okay if you use this library. If you do not know Python, it may be easier to quickly learn Python (watch a few tutorials) and use the CSP library than struggle using another language. However, if you are a very good Java or C++ programmer and you fully understand CSP solvers, this may not be that difficult to implement in those languages.

SUBMISSION:
Please submit all of the source code for your assignment along with your code walk-through video.