## Assignment #3 Problem Solving Due: Sunday, 10/08/17, 11:59pm

(30 pts) Let's use a game to teach algorithmic thinking. Go to http://robozzle.com.

- After you understand the tutorial, go to the EasytoHard tab to see a list of games to play.
- Under the tab that says EasytoHard, **Pick 3 games to play**. For each problem:
  - Write down the name and steps needed to solve each of these puzzles.
    - $\circ~$  Run the games to make sure your solution works.
    - Are there many solutions to any of the puzzles?

(30 pts) Complete the 3 of the 4 problem-solving steps to solve the following problem. Find the largest number out of 5 random numbers given to you from a friend. Follow the form outlined as follows:

## • Step 1: Problem Analysis.

- a. Comments about the problem to aid in understanding it.
- b. Description of the knowledge base (this list would include what you would be expected to know to follow the solution).
- c. What are the various solutions? Weigh pros and cons and pick one.
- Step 2: Program Design. List the specific steps that would enable another person to find the largest among the 5 numbers presented. Remember, you have to be very explicit here to make sure one can actually accomplish the task using your directions.
  - 1.
  - 2.
  - <u>2</u>. 3.
  - 3.

## • Step 4: Program Testing.

Create a Test Plan with several test cases including the average and extreme cases.

(10 pts) How does your solution/program design change if the size of the list is 50, 500, or n, instead of 5?

(30 pts) On a piece of paper, use your Lab #2 knowledge to write the Python3 syntax that would correspond to the algorithm you specified in Assignment #2 for converting a positive number less than 256 to binary. You can now use variables and built-in functions, such as input(), print(), etc., to interact with memory, the keyboard, and the monitor. You do not need to use anything that was not covered in Lab #2. **Challenge yourself and see how much you can actually code/implement in Python. (If you are more advanced, try doing this without using knowledge of conditions and loops, since that material has not been covered!).** 

- Were you explicit enough in your Assignment #2 algorithm?
- What steps did you have to add or delete to convert it to python syntax?

• What steps/part of your algorithm did you have to alter because you did not know what the corresponding python syntax would be?

## Extra Credit (5 pts):

Without using built-in functions, write the steps needed to provide an error message to the user when bad input from your test cases is entered, e.g. -2 prints a bad input message, t prints a bad input message, 2.3 prints a bad input message, etc. **Challenge yourself and try to code/implement this in Python without using built-in functions!!!** 

Electronically submit your document **as a pdf** by the assignment due date, using TEACH: <u>https://secure.engr.oregonstate.edu:8000/teach.php?type=want\_auth</u>