Exercise #1 (No computers needed)
Due Friday, 4/03/2015, at 11:59pm

(5 pts) Pointer Review
Each of the following declarations and program segments has errors. Locate as many as you can in A-M, and describe why it is an error.

A) int ptr*;
B) int x, *ptr;
   &x = ptr;
C) int x, *ptr;
   *ptr = &x;
D) int x, *ptr;
   ptr = &x;
   ptr = 100; // Store 100 in x
   cout << x << endl;
E) int numbers[] = {10, 20, 30, 40, 50};
   cout << "The third element in the array is ";
   cout << *numbers + 3 << endl;
F) int values[20], *iptr;
   iptr = values;
   iptr *= 2;
G) double level;
   int dPtr = &level;
H) int *iptr = &ivalue;
   int ivalue;
I) int *pint;
   new pint;
J) void doubleVal(int val) {
   *val *= 2;
}
K) int *pint;
pint = new int;
pint = 100;
L) int *pint;
pint = new int[100]; // Allocate memory
...
//Process the array
...
delete pint; // Free memory
M) int *getNum()
{
    int wholeNum;
    cout << "Enter a number: ";
    cin >> wholeNum;
    return &wholeNum;
}

(5 pts) Design and Testing
These are well-documented problem-solving steps outlined by George Polya used in mathematics, but we can adapt these steps to computer science too. Three of these steps include:

Understanding the Problem
In your own words, explain what YOU think the problem is asking you to do. In this section, document your uncertainties about the problem and anything else that you feel was unclear or vague. This is to ensure that YOUR understanding matches MY understanding of the problem.

Devising a Plan/Design
At a minimum, provide an algorithm/pseudo code you designed to help solve the problem. In addition, include pictures/flow charts you used to help you devise your plan, as well as any other design decisions you made such as how to manage your time, how to decompose the problem, where to start first, etc. You can scan any handwritten work and attach it to the document as needed.
**Testing**
Report any checking/self-reflection you did while solving the problem. For instance, how did you make sense of the output from the implementation? This includes things such as using a calculator to make sure the output is correct, testing to make sure your code executes correctly and behaves the way you expect under specific circumstances, using external sources of information such as the internet to make sense of the results, etc. In addition, you will **provide us a test plan!**

Please see the template for this document: [Polya_template.pdf](#)

**Using Assignment #1, as a group, answer the following questions:**

**Understanding the Problem** – Do you know how to play connect four? Do you understand how to play connect any?

**Design** – What will be the function calls for these prototypes? Write the flow-chart and/or pseudo-code for these function definitions.

```c
bool is_valid_arguments(char *info[]);
void set_game_info(game *, char *info[]);
char** create_table(int, int);
void play_game(game *);
bool check_winner(game);
void delete_table(game *);
```

**Testing** – Create a test plan with the test cases (bad, good, and edge cases). What are the expected results?