CSI201 – Hangman Game

Practical:

For this assignment, you will write a hangman game. Your program will randomly pick one from a list of possible words. The user/player will then attempt to guess the letters within that word, and be shown only the letters that they have correctly guessed so far. If the player can correctly guess all of the letters in this word before making six incorrect guesses, they win. Otherwise they lose. Below are some tips to consider as you develop this program.

1. Create an empty new project in Visual Studio called YourName\_Assmt4 with a single C++ file. Fill this file with C++ code to implement this game. It may be easiest to implement this entire game within the main function, so please feel free to do so.
2. Your list of random words can be a two-dimensional array of characters, where the first index identifies a single word in the list, and the second index identifies a single character position within that word. It will be easiest to make all of these words the same length. After you get this working, you are welcome to try generalizing your code to handle words of different lengths. Here’s an example of some code that prints a random word from a two-dimensional array of characters:

 // 2d array of characters containing animal names

 char animals[ANIMAL\_COUNT][NAME\_LENGTH] = {

 "rabbit",

 "monkey",

 "jaguar",

 "turtle",

 "dragon"

 };

 // generate a random index 0 to 4 (less than ANIMAL\_COUNT)

 srand(time(0)); rand();

 int randomAnimal = rand() % ANIMAL\_COUNT;

 // print the name of the randomly selected animal

 cout << animals[randomAnimal] << endl;

Notice that this code relies on the following preprocessor directives being set outside/above the main function’s definition:

#include <ctime>

#define ANIMAL\_COUNT 5

#define NAME\_LENGTH 7

1. Another tricky part of this game is thinking about how you will track and display the portion of the mystery word that the player has guessed. It will probably be easiest to create and use a separate character array for this purpose. This array can start out filled with asterisks that represent blanks. Then as players guess letters that are contained within the mystery word, you can copy these letters from the two-dimensional array with all of the words into this single dimensional array that can be easily drawn to the screen for the player.
2. The guts of implementing this game will be to 1) determine whether the letter that was guessed is a part of the randomly chosen word, and then either 2) copy that letter into the corresponding position of the visible array, or 3) track progress towards losing the games. After each guess, you will also need to check whether the game has been won: by the player correctly guessing every letter in the mystery word, or lost: by the player making their sixth incorrect guess.

# Problems:

1. Write a line of code to create a new array of twelve integers called weights:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Write a line of code to set the third variable in this weights array to the value six (remember that the index of the first variable in any array is zero):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Write a for loop that prints all twelve values in the weights array to the console:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Write a line of code to create a new two-dimensional array of integers called newWeights, with dimensions of size three and four:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Finally, write a line of code that sets the second variable in the third array of this two-dimensional newWeights array to the value five:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submission:

**Name:** [REPLACE WITH YOUR NAME]

**Honor Code:** [REPLACE WITH YOUR HONOR CODE PLEDGE]

**Resources:** [REPLACE WITH NAMES OF PEOPLE, WEBSITES, AND OTHER RESOURCES USED, ALONG WITH A BRIEF DECSRIPTION OF THE KIND OF HELP THEY PROVIDED]

After you have completed the practical portion of this assignment, zip your project folder into a single file and upload it to Canvas. After you have completed the written problems above, save this document (with your answers), and then upload it to canvas. For full credit, you will need to submit both halves of this assignment on Canvas, no later than **Thursday April 2nd, 2015**.