CSI201 – Guessing Game

Practical:

For this assignment, you will write a high/low guessing game that is implemented as a recursive function. The player will start by guessing the mysteryNumber that is randomly assigned a value between 1 and 100. When the player guesses, this range will contract. For example, if I guess 72 when the mystery number is 18, I will be told that my guess is too high and the range for my next guess will contract to between 1 and 71. Here is the order that I would suggest building this program in:

1. Create an empty new project in Visual Studio called *YourName*\_Assmt3 with a single C++ file. Fill this file with C++ code that prints the following text message to the screen: *Guesing Game 1.0*.
2. Next write a function called readGuess with the following signature: int readGuess(int min, int max). This function should prompt the player to enter their next guess by showing them the current range of possible values: min to max. The number that this function reads in from the player should then be returned as output from this function.
3. After you have tested this function to ensure that it works, replace your main function with the following code:

void main()

{

// generates a random mysteryNumber: 1-100

int min = 1;

int max = 100;

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

int mysteryNumber = rand()%(max-min+1)+min;

// plays the entire high/low guessing game

highLowGame(mysteryNumber,min, max);

}

1. Next you will implement the highLowGame function that is being called from the main function defined above. Note that you should not change this main function definition to complete the assignment.
2. Your highLowGame function should start by calling your readGuess function to determine the player’s next guess.
3. Once you have the player’s next guess, you will need to compare that guess to the mysteryNumber to determine and display whether the player’s guess is too large, too small, or exactly the same.
4. When the mysteryNumber and player’s guess match, the highLowGame function should display a message saying this, and then return.
5. When the mysteryNumber and player’s guess are different, the highLowGame function should display a message saying this, and it should then make a recursive call to itself with an adjusted min or max value that reflects the contracted range of possible mystery number values.

# Problems:

Consider the following code to answer the written questions below.

#include <iostream>

using namespace std;

int a(int a1);

int b(int b1, int b2);

int c(int c1, int c2);

void main()

{

cout << a(2) + b(3,4) << endl;

}

int a(int a1)

{

if(a1 < 0) return a1;

return a(a1 - 7);

}

int b(int b1, int b2)

{

if(b1 > b2) return b1;

return c(b2,b1+1);

}

int c(int c1, int c2)

{

if(c1 < c2) return c1;

return b(c2,c1-1);

}

1. How many functions are defined above? \_\_\_\_\_\_\_\_\_\_\_\_

2. Circle each of these definitions above, or highlight them in yellow.

3. How many functions are declared above? \_\_\_\_\_\_\_\_\_\_\_\_

4. Put a box around each of these declarations, or highlight them in blue.

5. When this program runs, how many times is the a-function evaluated? \_\_\_\_\_\_\_\_\_\_\_\_

6. When this program runs, how many times is the b-function evaluated? \_\_\_\_\_\_\_\_\_\_\_\_

7. What output will be printed to the screen when this program is run? \_\_\_\_\_\_\_\_\_\_\_\_

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 **Tuesday March 3rd, 2015**.