Credit Name: "Credit Name: CSE 2140 2nd Language Programming
Assignment Name: Chapter 3 Portfolio

Error Log Entry

What error message did you encounter (if any)?
I had the most difficulty trying to figure out how to make the colors and number of pegs be outputted properly.
What unexpected behavior did your program exhibit?
1. When I first tested my program, I realized I was using the same value for number of pegs and colors. This is caused issues because when the user chose a for example: higher number of colors rather than pegs, the program would just rely how many pegs colors are right, without giving the accurate track which colors appeared in the secret code.
2. As I was trying to fix my code, I also realized that I had misunderstood the objective of the exercise by accident. I was supposed to make it so that after each guess the program would ouput the number of pegs who had the correct colors in the correct position and the colors that were in the secret code regardless of position. I had coded my program to count only the colors in the secret code but in the wrong position. This then made it so that when the u
What caused the issue? (e.g., syntax error, logic error, incorrect function usage, etc.)
1. This was a logic error as I accidentally made it so the result would display the same as the pegs.
2. This was a logic error on my part since I didn't read properly. But I just re did my code for the colors output.
Include a screenshot of specific lines of code.

```
    // Display results
    System.out.println(
        "You have " + result[0] + " peg(s) correct and " +
        result[0] + " color(s) correct."
    );
```

```
mastermind tester [Java Application] C:\Program Files\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32
Enter the number of pegs <1-10>:
3
Enter the number of colors <1-9>:
4
Guess1:
Color for peg 1: 1
Color for peg 2: 2
Color for peg 3: 3
You have 1 peg(s) correct and 1 color(s) correct.
Guess2:
Color for peg 1: 2
Color for peg 2: 3
Color for peg 3: 4
You have 1 peg(s) correct and 1 color(s) correct.
Guess3:
Color for peg 1:
```

```
Enter the number of pegs <1-10>:
2
Enter the number of colors <1-9>:
2
Guess1:
Color for peg 1: 1
Color for peg 2: 2
You have 0 peg(s) correct and 2 color(s) correct.
Guess2:
Color for peg 1: 2
Color for peg 2: 1
You have 2 peg(s) correct and 0 color(s) correct.

You have broken the code in 2 guesses.
```

How did you fix the issue?
1. I put in the array the correct number that repersented the correct varaible in which the colors would be repersented in. And I made sure to include both the colors in the right positiona and the ones in who are there but wrong position.
2.For this, I tweaked a few things and made sure to rename my variables so I knew what contained what value.

Provide the corrected code or solution using a screenshot.

```
int guessNum = 1;
    // while loop for game
    while (true) {
        // Guess#
        System.out.println("Guess" + guessNum + ":");
        int[] guess = new int[numPegs];

        // Read the user's guess
        for (int i = 0; i < numPegs; i++) {
            System.out.print("Color for peg " + (i + 1) + ": ");
            guess[i] = input.nextInt();
        }

        // Check the guess
        int[] result = game.checkGuess(guess);

        // Display results
        System.out.println(
            "You have " + result[0] + " peg(s) correct and " +
            (result[1]+ result[0]) + " color(s) correct."
        );

        // Check win condition
        if (result[0] == numPegs) {
            System.out.println(
                "\nYou have broken the code in " + guessNum + " guesses."
            );
            break;
        }

        guessNum++;
    }
    }
```

```
Enter the number of pegs <1-10>:
2
Enter the number of colors <1-9>:
2
Guess1:
Color for peg 1: 1
Color for peg 2: 2
```

```
    //
    public mastermindGame(int numPegs, int numColors) {
        this.numColors= numColors;
        this.numPegs= numPegs;
        secretcode = new int [numPegs];

        Random rand = new Random();
        for (int i = 0; i < numPegs; i++) {
            secretcode[i] = rand.nextInt(numColors) +1;


        }
    }
public int[] checkGuess(int[] guess) {
    int colorsAndpegs = 0;
    int colors = 0;

    int[] secretCount = new int[numColors + 1];
    int[] guessCount = new int[numColors + 1];

    for (int i = 0; i < numPegs; i++) {
        // code is cracked
        if (guess[i] == secretcode[i]) {
            colorsAndpegs++;
        } else {
            // wrong code --- adding of guess
            secretCount[secretcode[i]]++;
            guessCount[guess[i]]++;
        }
    }
    for (int c = 1; c <= numColors; c++) {
        if (secretCount[c] < guessCount[c]) {
            colors += secretCount[c];
        } else {
            colors += guessCount[c];
        }
    }
    return new int[] {colorsAndpegs, colors};
```

```
You have 1 peg(s) correct and 1 color(s) correct.
Guess2:
Color for peg 1: 2
Color for peg 2: 2
You have 0 peg(s) correct and 0 color(s) correct.
Guess3:
Color for peg 1: 1
Color for peg 2: 1
You have 2 peg(s) correct and 2 color(s) correct.

You have broken the code in 3 guesses.
```

FULL CODE + TEST

```
import java.util.Random;
// Introducing your variables ---
public class mastermindGame {
    private int[] secretcode;
    private int numPegs;
    private int numColors;

    public mastermindGame(int numPegs, int numColors) {
        this.numColors= numColors;
        this.numPegs= numPegs;
        secretcode = new int [numPegs];
        // randomizer
        Random rand = new Random();
        for (int i = 0; i < numPegs; i++) {
            secretcode[i] = rand.nextInt(numColors) +1;


        }
    }
public int[] checkGuess(int[] guess) {
    int colorsAndpegs = 0; //the correct code-> correct color and position
    int colors = 0; // the colors regardless of position

    //creating array's --> index starts at 1

    int[] secretCount = new int[numColors + 1]; //how many times each color in the code
    int[] guessCount = new int[numColors + 1]; // how many times each color appears in the guess

    for (int i = 0; i < numPegs; i++) {
        // check if guessed color matches the secret code
        if (guess[i] == secretcode[i]) { //if code is correct
            colorsAndpegs++;
        } else {
            // wrong code --- adding of guess
            secretCount[secretcode[i]]++; // if code is not correct
            guessCount[guess[i]]++;
        }
    }
    for (int c = 1; c <= numColors; c++) {
        if (secretCount[c] < guessCount[c]) {
            colors += secretCount[c];
        } else {
            colors += guessCount[c];
        }
    }
    return new int[] {colorsAndpegs, colors};
}


    }
```

```
package Mastery;

import java.util.Scanner;

public class mastermindTester {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner input = new Scanner(System.in);
        //asking the user how many pegs and colors + getting their answers
        System.out.println("Enter the number of pegs <1-10>: ");
        int numPegs = input.nextInt();
        System.out.println("Enter the number of colors <1-9>: ");
        int numColors = input.nextInt();

        // creating a new game
         mastermindGame game = new mastermindGame(numPegs, numColors);
         // introducing our guess variable
       int guessNum = 1;
            // while loop for game
            while (true) {
                // Guess#
                System.out.println("Guess" + guessNum + ":");
                int[] guess = new int[numPegs];

                // Read the user's guess
                for (int i = 0; i < numPegs; i++) {
                    System.out.print("Color for peg " + (i + 1) + ": ");
                    guess[i] = input.nextInt();
                }

                // Check the guess
                int[] result = game.checkGuess(guess);

                // Display results
                System.out.println(
                        "You have " + result[0] + " peg(s) correct and " +
                        result[1] + result[0]) + " color(s) correct."
                );

                // Check win condition
                if (result[0] == numPegs) {
                    System.out.println(
                            "\nYou have broken the code in " + guessNum + " guesses."

                    );
                    break;
                }

                guessNum++;
            }
            }
    }
```

```
Enter the number of pegs <1-10>:
3
Enter the number of colors <1-9>:
3
Guess1:
Color for peg 1: 1
Color for peg 2: 2
Color for peg 3: 3
You have 2 peg(s) correct and 2 color(s) correct.
Guess2:
Color for peg 1: 1
Color for peg 2: 2
Color for peg 3: 1
You have 1 peg(s) correct and 1 color(s) correct.
Guess3:
Color for peg 1:
2
Color for peg 2: 2
Color for peg 3: 1
You have 2 peg(s) correct and 2 color(s) correct.
Guess4:
Color for peg 1: 2
Color for peg 2: 2
Color for peg 3: 3
You have 3 peg(s) correct and 3 color(s) correct.

You have broken the code in 4 guesses.
```