Operation #7
CS 1400 – Satterthwaite

Goal: This assignment will have you learn how to work with arrays and loops.

Assignment: Create a modified Yahtzee game **AND NAME IT Yahtzee**. This assignment requires that you implement 5 dice using arrays. Assignments not using arrays will receive no credit.

The object of this game is to randomly roll 5 dice. Display the results to the user. Let the user choose to reroll any number of the dice. (For example, if the user chooses zero, then no dice will be re-rolled. If 1, 2, 3, and 5 are chosen, then all dice except for die #4 will be rerolled. I am open to various approaches you take to let the user select how to reroll. Just know that first the user must indicate which dice to roll or which to hold. And only then once all are entered does a full reroll take place. The easiest way to do this is simply ask five questions, one for each dice.)

After two additional re-rolls (for 3 rolls in total), the dice no longer needs to be rolled.

For the sake of grading and for testing, provide a special "cheater mode" in your code after the 3 rolls. The cheater mode should ask the user to input 5 new integers from 1-6. Those integers will become the new dice values. (For example, a user can enter "3 3 3 3 3" to change all the dice to showing the value 3, thus ensuring a Yahtzee. But the easiest way to do this is simply ask five questions, one for each dice.)

Once this process is completed, compute the best possible winning scenario for the dice. For example, if a user rolled "3 3 3 3 3", you could potentially display a Yahtzee, a four of a kind, a three of a kind, and a full house, as all technically apply. But you should only display that it is a Yahtzee. The order in rankings go Yahtzee (five of the same kind), large straight (five in a row), small straight (a four in a row exists), four of a kind, full house (two of a kind and a three of a kind), three of a kind. You do NOT need to compute a point value score. You only need to display what kind of scenario it is.

Although the game would be fun and more practical to run for multiple rounds, I don't want to overwhelm you for this assignment. Your program then needs only to run for one total round.

Please note that the starter code gives you a considerable head start and a great approach to determining which scenarios the dice fall into. If you weren't in class, watch the lecture video for this.

```
package insertPackageNameHere;

import java.util.Random;
```

```
public class InsertClassNameHere {

    public static void main(String[] args) {

        int[] dice = new int[5];
        Random r = new Random();

        //generate the dice rolls
        for (int i = 0; i < 5; i++) {
            dice[i] = r.nextInt(6) + 1;
        }

        //display the dice rolls
        for (int i = 0; i < 5; i++) {
            System.out.print(dice[i] + " ");

        }
        System.out.println();


        //TODO: set up two rounds of rerolling


        //Create a counts array, set all values to zero.
        int[] counts = new int[6];
        for (int i = 0; i < 6; i++){
            counts[i] = 0;
        }
        //count up the values
        for (int i = 0; i < 5; i++) {
            /* diceIndex will hold the value of the dice location
               minus one.  This is so it can be placed in the 0th
               based array, in the correct location.*/
            int diceIndex = dice[i] - 1;
            /* This will go to the location reserved for the dice and
               increment its value by 1.
               For example, if the value of the die was 4, the
               diceIndex value would be 3.  This would increment
               counts[3] by one. Now it’s easy to tell how many 4s
               there are, because each time there is a 4, it will
               increment the count[3] location by one again.*/
            counts[diceIndex]++;
        }
```

```
          /* This prints out how many of each number there is.
          COMMENT IT OUT WHEN YOU ARE DONE WITH IT. */
          System.out.println();
          for (int i = 0; i < 6; i++){
            System.out.println("The number of " + (i+1) + "s is : " +
counts[i]);
        }

    }
}
```