Yahtzee!

Based on the assignment idea of Eric Roberts and Julie Zelenski, Stanford University

For this assignment, you will be implementing the classic dice game of Yahtzee. For those who are not familiar with it, Yahtzee is a game played with five dice. On a players turn, they roll their dice, and then have up to two additional rolls in which they may reroll any number of the dice. After the rerolls, the player chooses the category in which to score the roll.

Dice categories

The thirteen categories of dice configurations and their scores are:

1. Ones. Any dice configuration is valid for this category. The score is equal to the sum of all of the 1's showing on the dice, which is 0 if there are no 1's showing.

2–6. Twos, Threes, Fours, Fives, and Sixes. (Similar to above but for different values). Any dice configuration is valid for these categories. The score is equal to the sum of the 2's, 3's, 4's, 5's, or 6's, showing on the dice.

7. Three of a Kind. At least three of the dice must show the same value. The score is equal to the sum of all of the values showing on the dice.

8. Four of a Kind. At least four of the dice must show the same value. The score is equal to the sum of all of the values showing on the dice.

9. Full House. The dice must show three of one value and two of another value (Special case: a five-of-a-kind also counts as a full house). The score is 25 points.

10. Small Straight. The dice must contain at least four consecutive values, such as the sequence 2-3-4-5. The score is 30 points.

11. Large Straight. The dice must contain five consecutive values, such as the sequence 1-2-3-4-5. The score is 40 points.

12. Yahtzee! All of the dice must show the same value. The score is 50 points.

13. Chance. Any dice configuration is valid for this category. The score is equal to the sum of the values showing on the dice.

Your Task

For this assignment, we have created a starter file with most of the game engine already built for you. The only thing missing is the thirteen scoring functions. You'll notice that in the main there is a switch statement which is commented out. It is commented out because it calls functions which have not yet been defined. Your job is to define those thirteen functions, each of which calculates the score for a given category. Here's an example skeleton of one of the functions:

```
int scoreOnes(int ones, int twos, int threes, int fours, int fives,
              int sixes)

{

}
```

In main, the rolls are tallied by value, so that rather than working with the raw dice, you'll be using integers which tell you how many dice are present with each possible value. For categories where the score is all or nothing (such as the small and large straights), the function should return the full value if the dice match the requirements, and zero if they do not (sometimes you may be forced to assign a die roll to a non-matching category later in the game).

In getting started, it may be helpful to first write stubs for each of the functions. A stub looks something like this:

```
int scoreOnes(int ones, int twos, int threes, int fours, int fives,
              int sixes)

{

     return 0;

}
```

Since the function always returns zero, it of course isn't providing the correct answer, but it does allow the program to compile so you can test your work. Once you've got the stubs in place, you can fill out the functions with a more meaningful implementation one at a time, so you can test your work little by little instead of all at once at the end.

Additionally, a number of categories make use of the sum of the dice, so you may wish to define a helper function which calculates that sum. This will allow you to simply call that function as needed instead of rewriting the same code every time you need that calculation.

Your final submission should simply be the main.cpp file provided with your functions added. Be sure to comment your functions as well!