Daniel Nedrow

CEG 3110 Project 3

**Requirements**

(1) When scoring the Aces category, the program shall sum only dice with the value “1” and assign that sum as the score for Aces. If “1” is not shown on any dice, the program shall assign a zero for Aces.

(2) When scoring the Twos category, the program shall sum only dice with the value “2” and assign that sum as the score for Twos. If “2” is not shown on any dice, the program shall assign a zero for Twos.

(3) When scoring the Threes category, the program shall sum only dice with the value “3” and assign that sum as the score for Threes. If “3” is not shown on any dice, the program shall assign a zero for Threes.

(4) When scoring the Fours category, the program shall sum only dice with the value “4” and assign that sum as the score for Fours. If “4” is not shown on any dice, the program shall assign a zero for Fours.

(5) When scoring the Fives category, the program shall sum only dice with the value “5” and assign that sum as the score for Fives. If “5” is not shown on any dice, the program shall assign a zero for Fives.

(6) When scoring the Sixes category, the program shall sum only dice with the value “6” and assign that sum as the score for Sixes. If “6” is not shown on any dice, the program shall assign a zero for Sixes.

(7) When scoring the 3-of-a-kind category, the program shall first check that three or more dice match the same value. If this is the case, the score shall be the total of all five dice; if not, the score shall be zero.

(8) When scoring the 4-of-a-kind category, the program shall first check that four or more dice match the same value. If this is the case, the score shall be the total of all five dice; if not, the score shall be zero.

(9) When scoring the Full House category, the program shall first check that exactly three of the dice match each other with one value, and that the other two dice match each other with another value (the values for the group of three dice and the group of two dice may be the same as well; i.e., 5-of-a-kind is acceptable). If this is the case, the score shall be 25; if not, the score shall be zero.

(10) When scoring the Small Straight category, the program shall check that at least four of the dice have distinct values which can be arranged into consecutive order (1-2-3-4, 2-3-4-5, or 3-4-5-6, with the 5th die being irrelevant in all cases). If this is the case, the score shall be 30; if not, the score shall be zero.

(11) When scoring the Large Straight category, the program shall check that all five of the dice have distinct values which can be arranged into consecutive order (1-2-3-4-5 or 2-3-4-5-6). If this is the case, the score shall be 40; if not, the score shall be zero.

(12) When scoring the YAHTZEE category, the program shall check that all five dice match the same value. If this is the case, the score shall be 50; if not, the score shall be zero.

(13) When scoring the Chance category, the program shall sum all five dice.

(14) When scoring any category other than Yahtzee itself, the program shall check for a bonus Yahtzee using the following criteria: all five dice must match the same value, and the regular Yahtzee category must have already been scored with 50 points (and not with 0 points). Whenever this is the case, 100 additional points shall be added into Yahtzee Bonus. If these conditions don’t hold, no Yahtzee Bonus shall be added.

(15) When the game is complete, the program shall subtotal the upper section by adding the scores of each individual category (Aces through Sixes) of the upper section. This subtotal shall be displayed as Upper Section Subtotal (The Yahtzee game sheet uses the term “Total Score,” which is misleading).

(16) When the game is complete, the program shall add 35 to the upper section bonus score if and only if the player scores 63 points or more in the upper section subtotal. This score shall be displayed as Bonus.

(17) When the game is complete, the program shall calculate the upper section total as the sum of the upper section subtotal and the upper section bonus (if any). This score shall be displayed as Total of Upper Section.

(18) When the game is complete, the program shall display the total of the Yahtzee Bonus category (which was being tracked per requirement 14).

(19) When the game is complete, the program shall subtotal the lower section by adding the scores of each individual category (3-of-a-Kind through Yahtzee Bonus) of the lower section. This score shall be displayed as Total of Lower Section.

(20) When the game is complete, the program shall restate the Total of Upper Section.

(21) When the game is complete, the program shall calculate the grand total as the sum of the Total of Lower Section and the Total of Upper Section. This score shall be displayed as the Grand Total.

**Test Plan**

The test cases will be broken into four sets representing four separate, complete Yahtzee games. Each game will have a total of 15 test cases: 13 for scoring each individual category, and one each for scoring the Upper Section and the Lower Section (which will include the Grand Total). Before presenting the test cases for each game, I will briefly discuss some of the things that are meant to be tested by that game. In order to maintain the context of a game, it is important that the test cases for that game be run in order (for correct handling of Yahtzee Bonuses).

**Game A**

Game A tests scoring a zero in every category but Chance. Since there is no way to get a “near miss” on individual Upper Section categories, the tests for these categories are somewhat redundant. However, every Lower Section category (except Chance) will be tested with a near miss. For example, only three matching dice are rolled when scoring 4-of-a-Kind, only four dice that can be arranged in consecutive order are rolled when scoring Large Straight, and so on. This will exercise the boundaries of these categories. This game will also test the most extreme case of the Upper Section Bonus: obviously no Bonus should be awarded.

Another important test in Game A is what happens when 5-of-a-kind is rolled after Yahtzee has already been scored as a zero: in this case, no Yahtzee Bonus is awarded. I will test this immediately in the first two test cases (note absence of Yahtzee Bonus in Test Case 2a).

|  |  |  |
| --- | --- | --- |
| **Test Case 1a:** Yahtzee  Input: 4-4-5-4-4, Yahtzee  Expected Output:  0 points for Yahtzee  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 2a:** Chance  Input: 5-5-5-5-5, Chance  Expected Output:  25 points for Chance  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 3a:** Aces  Input: 2-2-3-3-4, Aces  Expected Output:  0 points for Aces  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 4a:** Twos  Input: 1-6-3-3-4, Twos  Expected Output:  0 points for Twos  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 5a:** Threes  Input: 2-2-6-5-4, Threes  Expected Output:  0 points for Threes  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 6a:** Fours  Input: 2-2-3-3-5, Fours  Expected Output:  0 points for Fours  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 7a:** Fives  Input: 2-2-3-3-4, Fives  Expected Output:  0 points for Fives  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 8a:** Sixes  Input: 2-2-3-3-4, Sixes  Expected Output:  0 points for Sixes  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 9a:** 3-of-a-Kind  Input: 1-3-5-4-4, 3-of-a-Kind  Expected Output:  0 points for 3-of-a-Kind  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 10a:** 4-of-a-Kind  Input: 1-4-5-4-4, 4-of-a-Kind  Expected Output:  0 points for 4-of-a-Kind  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 11a:** Full House  Input: 1-1-1-1-2, Full House  Expected Output:  0 points for Full House  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 12a:** Small Straight  Input: 1-1-2-2-3, Small Straight  Expected Output:  0 points for Small Straight  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 13a:** Large Straight  Input: 1-1-2-3-4, Large Straight  Expected Output:  0 points for Large Straight  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 14a:** Upper Scoring  Input: previous 13 test cases  Expected Output:  Upper Section Subtotal: 0  Bonus: 0  Total of Upper Section: 0  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 15a:** Lower Scoring  Input: previous 14 test cases  Expected Output:  Yahtzee Bonus: 0  Total of Lower Section: 25  Total of Upper Section: 0  Grand Total: 25  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |

**Game B**

Game B begins to test the boundary condition for scoring the Upper Section Bonus: we will miss the Bonus requirement by one point, so we expect a zero on the Bonus here. Necessarily, we will now actually be scoring some points in the Upper Section. We take the easiest way to get to 62: three qualifying dice each on the Twos through Sixes categories, but just two qualifying dice on the Aces category.

We will finish testing the boundary conditions of the Lower Section categories in the sense that we will qualify for each category with the exact minimum: exactly three matching dice for 3-of-a-Kind, exactly four dice which can be arranged in consecutive order for Small Straight, and so on. For Game B, we will test dice rolls which do not need to be sorted to see things like a Full House or a straight. The next game will test unsorted rolls. There are no potential Yahtzee Bonuses in this game, although merely rolling a single Yahtzee will test that the Yahtzee Bonus is not erroneously added onto the Yahtzee-scoring roll itself (this should never happen).

|  |  |  |
| --- | --- | --- |
| **Test Case 1b:** Aces  Input: 1-1-2-3-5, Aces  Expected Output:  2 points for Aces  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 2b:** Twos  Input: 2-2-2-6-5, Twos  Expected Output:  6 points for Twos  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 3b:** Threes  Input: 3-3-3-2-4, Threes  Expected Output:  9 points for Threes  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 4b:** Fours  Input: 4-4-4-2-3, Fours  Expected Output:  12 points for Fours  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 5b:** Fives  Input: 5-5-5-2-3, Fives  Expected Output:  15 points for Fives  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 6b:** Sixes  Input: 6-6-6-2-3, Sixes  Expected Output:  18 points for Sixes  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 7b:** 3-of-a-Kind  Input: 4-4-4-1-5, 3-of-a-Kind  Expected Output:  18 points for 3-of-a-Kind  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 8b:** 4-of-a-Kind  Input: 2-2-2-2-3, 4-of-a-Kind  Expected Output:  11 points for 4-of-a-Kind  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 9b:** Full House  Input: 2-2-2-1-1, Full House  Expected Output:  25 points for Full House  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 10b:** Small Straight  Input: 2-3-4-5-4, Small Straight  Expected Output:  30 points for Small Straight  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 11b:** Large Straight  Input: 1-2-3-4-5, Large Straight  Expected Output:  40 points for Large Straight  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 12b:** Yahtzee  Input: 1-1-1-1-1, Yahtzee  Expected Output:  50 points for Yahtzee  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 13b:** Chance  Input: 1-2-3-4-5, Chance  Expected Output:  15 points for Chance  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 14b:** Upper Scoring  Input: previous 13 test cases  Expected Output:  Upper Section Subtotal: 62  Bonus: 0  Total of Upper Section: 62  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 15b:** Lower Scoring  Input: previous 14 test cases  Expected Output:  Yahtzee Bonus: 0  Total of Lower Section: 189  Total of Upper Section: 62  Grand Total: 251  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |

**Game C**

Game C will finish testing the boundary condition for scoring the Upper Section Bonus: we will exactly match the minimum Upper Section score required for the Bonus, so we expect a 35 on the Bonus here. For variety in our test cases, we will refrain from taking the easiest way to get to 63 (three qualifying dice in each category). We take an alternate route.

We will test exceeding the minimum requirements for 3-of-a-Kind (by rolling four matching dice) and Small Straight (by rolling five dice which can be arranged in consecutive order). These categories should still score normally. We cannot exceed 4-of-a-Kind or Full House without also testing Yahtzee Bonuses, and those tests will be saved for our final game. Of course, the requirements for Large Straight and Yahtzee cannot be exceeded, and Chance has no requirements to exceed.

As promised, we will now take care to test unsorted rolls (except of course for Yahtzee). As with the last game, there will be no potential Yahtzee Bonuses in this game.

|  |  |  |
| --- | --- | --- |
| **Test Case 1c:** Aces  Input: 1-5-1-3-1, Aces  Expected Output:  3 points for Aces  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 2c:** Twos  Input: 5-6-4-3-2, Twos  Expected Output:  2 points for Twos  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 3c:** Threes  Input: 2-4-3-5-3, Threes  Expected Output:  6 points for Threes  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 4c:** Fours  Input: 6-4-3-4-3, Fours  Expected Output:  8 points for Fours  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 5c:** Fives  Input: 5-5-3-5-5, Fives  Expected Output:  20 points for Fives  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 6c:** Sixes  Input: 6-6-6-1-6, Sixes  Expected Output:  24 points for Sixes  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 7c:** 3-of-a-Kind  Input: 6-1-6-6-6, 3-of-a-Kind  Expected Output:  25 points for 3-of-a-Kind  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 8c:** 4-of-a-Kind  Input: 5-5-1-5-5, 4-of-a-Kind  Expected Output:  21 points for 4-of-a-Kind  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 9c:** Full House  Input: 6-5-5-6-5, Full House  Expected Output:  25 points for Full House  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 10c:** Small Straight  Input: 3-6-5-4-2, Small Straight  Expected Output:  30 points for Small Straight  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 11c:** Large Straight  Input: 5-2-6-3-4, Large Straight  Expected Output:  40 points for Large Straight  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 12c:** Yahtzee  Input: 6-6-6-6-6, Yahtzee  Expected Output:  50 points for Yahtzee  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 13c:** Chance  Input: 6-6-5-5-6, Chance  Expected Output:  28 points for Chance  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 14c:** Upper Scoring  Input: previous 13 test cases  Expected Output:  Upper Section Subtotal: 63  Bonus: 35  Total of Upper Section: 98  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 15c:** Lower Scoring  Input: previous 14 test cases  Expected Output:  Yahtzee Bonus: 0  Total of Lower Section: 219  Total of Upper Section: 98  Grand Total: 317  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |

**Game D**

The purpose of Game D is to thoroughly test the Yahtzee Bonus, and at the same time test how individual categories themselves score a roll of 5-of-a-kind (which we will roll for every category). It is important to note that the Small Straight and Large Straight are the only two categories which cannot be satisfied by rolling a 5-of-a-kind: we expect to score zero for these categories (but still obtain the Yahtzee Bonus if we score these after a Yahtzee). As discussed in class, the Full House definition allows for a 5-of-a-kind, so it is important that we get 25 points for Full House in this game. It is easy to see that the remaining categories can be satisfied by 5-of-a-kind. Yet for the sake of variety, two test cases for the Upper Section will score a zero: the dice will show 5-of-a-kind, but the wrong value for that category. Again, we expect to get the Yahtzee Bonus (if this category is scored after a Yahtzee) whether we are getting points in the home category or not.

It is worth repeating that these test cases must be run in the specified order: categories scored after the Yahtzee will be eligible for a Yahtzee Bonus, while categories scored before the Yahtzee will not. This is a critical part of testing the Yahtzee Bonus in this game. I tried to mix in a balance of Upper Section and Lower Section categories both before and after the Yahtzee, and there is a straight both before and after the Yahtzee. As such, the order of the categories looks a bit messier than in previous games, and this seems more realistic as to how a Yahtzee game would unfold (of course, rolling a 5-of-a-kind every time isn’t realistic!). This messy ordering will test that scoring can be done in an order other than what appears on the score sheet.

|  |  |  |
| --- | --- | --- |
| **Test Case 1d:** Fours  Input: 4-4-4-4-4, Fours  Expected Output:  20 points for Fours  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 2d:** Twos  Input: 3-3-3-3-3, Twos  Expected Output:  0 points for Twos  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 3d:** Sixes  Input: 6-6-6-6-6, Sixes  Expected Output:  30 points for Sixes  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 4d:** 3-of-a-Kind  Input: 5-5-5-5-5, 3-of-a-Kind  Expected Output:  25 points for 3-of-a-Kind  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 5d:** Small Straight  Input: 3-3-3-3-3, Small Straight  Expected Output:  0 points for Small Straight  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 6d:** Yahtzee  Input: 2-2-2-2-2, Yahtzee  Expected Output:  50 points for Yahtzee  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 7d:** Full House  Input: 1-1-1-1-1, Full House  Expected Output:  25 points for Full House  100 points for Yahtzee Bonus  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 8d:** Chance  Input: 6-6-6-6-6, Chance  Expected Output:  30 points for Chance  100 points for Yahtzee Bonus  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 9d:** 4-of-a-Kind  Input: 4-4-4-4-4, 4-of-a-Kind  Expected Output:  20 points for 4-of-a-Kind  100 points for Yahtzee Bonus  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 10d:** Aces  Input: 2-2-2-2-2, Aces  Expected Output:  0 points for Aces  100 points for Yahtzee Bonus  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 11d:** Large Straight  Input: 2-2-2-2-2, Large Straight  Expected Output:  0 points for Large Straight  100 points for Yahtzee Bonus  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 12d:** Threes  Input: 3-3-3-3-3, Threes  Expected Output:  15 points for Threes  100 points for Yahtzee Bonus  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |
| **Test Case 13d:** Fives  Input: 5-5-5-5-5, Fives  Expected Output:  25 points for Fives  100 points for Yahtzee Bonus  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 14d:** Upper Scoring  Input: previous 13 test cases  Expected Output:  Upper Section Subtotal: 90  Bonus: 35  Total of Upper Section: 125  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail | **Test Case 15d:** Lower Scoring  Input: previous 14 test cases  Expected Output:  Yahtzee Bonus: 700  Total of Lower Section: 850  Total of Upper Section: 125  Grand Total: 975  Actual Output:  Pass/Fail: [ ] Pass [ ] Fail |

**Output from Complete Automated Test Plan**

run:

---------------------------------------------

Game A:

Now running test 1a...

Dice values are: 4-4-5-4-4

Player selects Yahtzee

Sending input to Yahtzee program...

Expected Result: 0 points for Yahtzee

Actual Result: 0 points for Yahtzee

Results match -- test PASSES

Now running test 2a...

Dice values are: 5-5-5-5-5

Player selects Chance

Sending input to Yahtzee program...

Expected Result: 25 points for Chance

Actual Result: 25 points for Chance

Results match -- test PASSES

Now running test 3a...

Dice values are: 2-2-3-3-4

Player selects Aces

Sending input to Yahtzee program...

Expected Result: 0 points for Aces

Actual Result: 0 points for Aces

Results match -- test PASSES

Now running test 4a...

Dice values are: 1-6-3-3-4

Player selects Twos

Sending input to Yahtzee program...

Expected Result: 0 points for Twos

Actual Result: 0 points for Twos

Results match -- test PASSES

Now running test 5a...

Dice values are: 2-2-6-5-4

Player selects Threes

Sending input to Yahtzee program...

Expected Result: 0 points for Threes

Actual Result: 0 points for Threes

Results match -- test PASSES

Now running test 6a...

Dice values are: 2-2-3-3-5

Player selects Fours

Sending input to Yahtzee program...

Expected Result: 0 points for Fours

Actual Result: 0 points for Fours

Results match -- test PASSES

Now running test 7a...

Dice values are: 2-2-3-3-4

Player selects Fives

Sending input to Yahtzee program...

Expected Result: 0 points for Fives

Actual Result: 0 points for Fives

Results match -- test PASSES

Now running test 8a...

Dice values are: 2-2-3-3-4

Player selects Sixes

Sending input to Yahtzee program...

Expected Result: 0 points for Sixes

Actual Result: 0 points for Sixes

Results match -- test PASSES

Now running test 9a...

Dice values are: 1-3-5-4-4

Player selects 3-of-a-Kind

Sending input to Yahtzee program...

Expected Result: 0 points for 3-of-a-Kind

Actual Result: 0 points for 3-of-a-Kind

Results match -- test PASSES

Now running test 10a...

Dice values are: 1-4-5-4-4

Player selects 4-of-a-Kind

Sending input to Yahtzee program...

Expected Result: 0 points for 4-of-a-Kind

Actual Result: 0 points for 4-of-a-Kind

Results match -- test PASSES

Now running test 11a...

Dice values are: 1-1-1-1-2

Player selects Full House

Sending input to Yahtzee program...

Expected Result: 0 points for Full House

Actual Result: 0 points for Full House

Results match -- test PASSES

Now running test 12a...

Dice values are: 1-1-2-2-3

Player selects Small Straight

Sending input to Yahtzee program...

Expected Result: 0 points for Small Straight

Actual Result: 0 points for Small Straight

Results match -- test PASSES

Now running test 13a...

Dice values are: 1-1-2-3-4

Player selects Large Straight

Sending input to Yahtzee program...

Expected Result: 0 points for Large Straight

Actual Result: 0 points for Large Straight

Results match -- test PASSES

Now running test 14a...

Requesting Upper Scoring from Yahtzee program...

Expected Result:

Upper Section Subtotal: 0

Bonus: 0

Total of Upper Section: 0

Actual Result:

Upper Section Subtotal: 0

Bonus: 0

Total of Upper Section: 0

Results match -- test PASSES

Now running test 15a...

Requesting Lower Scoring from Yahtzee program...

Expected Result:

Yahtzee Bonus: 0

Total of Lower Section: 25

Total of Upper Section: 0

Grand Total: 25

Actual Result:

Yahtzee Bonus: 0

Total of Lower Section: 25

Total of Upper Section: 0

Grand Total: 25

Results match -- test PASSES

---------------------------------------------

Game B:

Now running test 1b...

Dice values are: 1-1-2-3-5

Player selects Aces

Sending input to Yahtzee program...

Expected Result: 2 points for Aces

Actual Result: 2 points for Aces

Results match -- test PASSES

Now running test 2b...

Dice values are: 2-2-2-6-5

Player selects Twos

Sending input to Yahtzee program...

Expected Result: 6 points for Twos

Actual Result: 6 points for Twos

Results match -- test PASSES

Now running test 3b...

Dice values are: 3-3-3-2-4

Player selects Threes

Sending input to Yahtzee program...

Expected Result: 9 points for Threes

Actual Result: 9 points for Threes

Results match -- test PASSES

Now running test 4b...

Dice values are: 4-4-4-2-3

Player selects Fours

Sending input to Yahtzee program...

Expected Result: 12 points for Fours

Actual Result: 12 points for Fours

Results match -- test PASSES

Now running test 5b...

Dice values are: 5-5-5-2-3

Player selects Fives

Sending input to Yahtzee program...

Expected Result: 15 points for Fives

Actual Result: 15 points for Fives

Results match -- test PASSES

Now running test 6b...

Dice values are: 6-6-6-2-3

Player selects Sixes

Sending input to Yahtzee program...

Expected Result: 18 points for Sixes

Actual Result: 18 points for Sixes

Results match -- test PASSES

Now running test 7b...

Dice values are: 4-4-4-1-5

Player selects 3-of-a-Kind

Sending input to Yahtzee program...

Expected Result: 18 points for 3-of-a-Kind

Actual Result: 18 points for 3-of-a-Kind

Results match -- test PASSES

Now running test 8b...

Dice values are: 2-2-2-2-3

Player selects 4-of-a-Kind

Sending input to Yahtzee program...

Expected Result: 11 points for 4-of-a-Kind

Actual Result: 11 points for 4-of-a-Kind

Results match -- test PASSES

Now running test 9b...

Dice values are: 2-2-2-1-1

Player selects Full House

Sending input to Yahtzee program...

Expected Result: 25 points for Full House

Actual Result: 25 points for Full House

Results match -- test PASSES

Now running test 10b...

Dice values are: 2-3-4-5-4

Player selects Small Straight

Sending input to Yahtzee program...

Expected Result: 30 points for Small Straight

Actual Result: 30 points for Small Straight

Results match -- test PASSES

Now running test 11b...

Dice values are: 1-2-3-4-5

Player selects Large Straight

Sending input to Yahtzee program...

Expected Result: 40 points for Large Straight

Actual Result: 40 points for Large Straight

Results match -- test PASSES

Now running test 12b...

Dice values are: 1-1-1-1-1

Player selects Yahtzee

Sending input to Yahtzee program...

Expected Result: 50 points for Yahtzee

Actual Result: 50 points for Yahtzee

Results match -- test PASSES

Now running test 13b...

Dice values are: 1-2-3-4-5

Player selects Chance

Sending input to Yahtzee program...

Expected Result: 15 points for Chance

Actual Result: 15 points for Chance

Results match -- test PASSES

Now running test 14b...

Requesting Upper Scoring from Yahtzee program...

Expected Result:

Upper Section Subtotal: 62

Bonus: 0

Total of Upper Section: 62

Actual Result:

Upper Section Subtotal: 62

Bonus: 0

Total of Upper Section: 62

Results match -- test PASSES

Now running test 15b...

Requesting Lower Scoring from Yahtzee program...

Expected Result:

Yahtzee Bonus: 0

Total of Lower Section: 189

Total of Upper Section: 62

Grand Total: 251

Actual Result:

Yahtzee Bonus: 0

Total of Lower Section: 189

Total of Upper Section: 62

Grand Total: 251

Results match -- test PASSES

---------------------------------------------

Game C:

Now running test 1c...

Dice values are: 1-5-1-3-1

Player selects Aces

Sending input to Yahtzee program...

Expected Result: 3 points for Aces

Actual Result: 3 points for Aces

Results match -- test PASSES

Now running test 2c...

Dice values are: 5-6-4-3-2

Player selects Twos

Sending input to Yahtzee program...

Expected Result: 2 points for Twos

Actual Result: 2 points for Twos

Results match -- test PASSES

Now running test 3c...

Dice values are: 2-4-3-5-3

Player selects Threes

Sending input to Yahtzee program...

Expected Result: 6 points for Threes

Actual Result: 6 points for Threes

Results match -- test PASSES

Now running test 4c...

Dice values are: 6-4-3-4-3

Player selects Fours

Sending input to Yahtzee program...

Expected Result: 8 points for Fours

Actual Result: 8 points for Fours

Results match -- test PASSES

Now running test 5c...

Dice values are: 5-5-3-5-5

Player selects Fives

Sending input to Yahtzee program...

Expected Result: 20 points for Fives

Actual Result: 20 points for Fives

Results match -- test PASSES

Now running test 6c...

Dice values are: 6-6-6-1-6

Player selects Sixes

Sending input to Yahtzee program...

Expected Result: 24 points for Sixes

Actual Result: 24 points for Sixes

Results match -- test PASSES

Now running test 7c...

Dice values are: 6-1-6-6-6

Player selects 3-of-a-Kind

Sending input to Yahtzee program...

Expected Result: 25 points for 3-of-a-Kind

Actual Result: 25 points for 3-of-a-Kind

Results match -- test PASSES

Now running test 8c...

Dice values are: 5-5-1-5-5

Player selects 4-of-a-Kind

Sending input to Yahtzee program...

Expected Result: 21 points for 4-of-a-Kind

Actual Result: 21 points for 4-of-a-Kind

Results match -- test PASSES

Now running test 9c...

Dice values are: 6-5-5-6-5

Player selects Full House

Sending input to Yahtzee program...

Expected Result: 25 points for Full House

Actual Result: 25 points for Full House

Results match -- test PASSES

Now running test 10c...

Dice values are: 3-6-5-4-2

Player selects Small Straight

Sending input to Yahtzee program...

Expected Result: 30 points for Small Straight

Actual Result: 30 points for Small Straight

Results match -- test PASSES

Now running test 11c...

Dice values are: 5-2-6-3-4

Player selects Large Straight

Sending input to Yahtzee program...

Expected Result: 40 points for Large Straight

Actual Result: 40 points for Large Straight

Results match -- test PASSES

Now running test 12c...

Dice values are: 6-6-6-6-6

Player selects Yahtzee

Sending input to Yahtzee program...

Expected Result: 50 points for Yahtzee

Actual Result: 50 points for Yahtzee

Results match -- test PASSES

Now running test 13c...

Dice values are: 6-6-5-5-6

Player selects Chance

Sending input to Yahtzee program...

Expected Result: 28 points for Chance

Actual Result: 28 points for Chance

Results match -- test PASSES

Now running test 14c...

Requesting Upper Scoring from Yahtzee program...

Expected Result:

Upper Section Subtotal: 63

Bonus: 35

Total of Upper Section: 98

Actual Result:

Upper Section Subtotal: 63

Bonus: 35

Total of Upper Section: 98

Results match -- test PASSES

Now running test 15c...

Requesting Lower Scoring from Yahtzee program...

Expected Result:

Yahtzee Bonus: 0

Total of Lower Section: 219

Total of Upper Section: 98

Grand Total: 317

Actual Result:

Yahtzee Bonus: 0

Total of Lower Section: 219

Total of Upper Section: 98

Grand Total: 317

Results match -- test PASSES

---------------------------------------------

Game D:

Now running test 1d...

Dice values are: 4-4-4-4-4

Player selects Fours

Sending input to Yahtzee program...

Expected Result: 20 points for Fours

Actual Result: 20 points for Fours

Results match -- test PASSES

Now running test 2d...

Dice values are: 3-3-3-3-3

Player selects Twos

Sending input to Yahtzee program...

Expected Result: 0 points for Twos

Actual Result: 0 points for Twos

Results match -- test PASSES

Now running test 3d...

Dice values are: 6-6-6-6-6

Player selects Sixes

Sending input to Yahtzee program...

Expected Result: 30 points for Sixes

Actual Result: 30 points for Sixes

Results match -- test PASSES

Now running test 4d...

Dice values are: 5-5-5-5-5

Player selects 3-of-a-Kind

Sending input to Yahtzee program...

Expected Result: 25 points for 3-of-a-Kind

Actual Result: 25 points for 3-of-a-Kind

Results match -- test PASSES

Now running test 5d...

Dice values are: 3-3-3-3-3

Player selects Small Straight

Sending input to Yahtzee program...

Expected Result: 0 points for Small Straight

Actual Result: 0 points for Small Straight

Results match -- test PASSES

Now running test 6d...

Dice values are: 2-2-2-2-2

Player selects Yahtzee

Sending input to Yahtzee program...

Expected Result: 50 points for Yahtzee

Actual Result: 50 points for Yahtzee

Results match -- test PASSES

Now running test 7d...

Dice values are: 1-1-1-1-1

Player selects Full House

Sending input to Yahtzee program...

Expected Result: 25 points for Full House

100 points for Yahtzee Bonus

Actual Result: 25 points for Full House

100 points for Yahtzee Bonus

Results match -- test PASSES

Now running test 8d...

Dice values are: 6-6-6-6-6

Player selects Chance

Sending input to Yahtzee program...

Expected Result: 30 points for Chance

100 points for Yahtzee Bonus

Actual Result: 30 points for Chance

100 points for Yahtzee Bonus

Results match -- test PASSES

Now running test 9d...

Dice values are: 4-4-4-4-4

Player selects 4-of-a-Kind

Sending input to Yahtzee program...

Expected Result: 20 points for 4-of-a-Kind

100 points for Yahtzee Bonus

Actual Result: 20 points for 4-of-a-Kind

100 points for Yahtzee Bonus

Results match -- test PASSES

Now running test 10d...

Dice values are: 2-2-2-2-2

Player selects Aces

Sending input to Yahtzee program...

Expected Result: 0 points for Aces

100 points for Yahtzee Bonus

Actual Result: 0 points for Aces

100 points for Yahtzee Bonus

Results match -- test PASSES

Now running test 11d...

Dice values are: 2-2-2-2-2

Player selects Large Straight

Sending input to Yahtzee program...

Expected Result: 0 points for Large Straight

100 points for Yahtzee Bonus

Actual Result: 0 points for Large Straight

100 points for Yahtzee Bonus

Results match -- test PASSES

Now running test 12d...

Dice values are: 3-3-3-3-3

Player selects Threes

Sending input to Yahtzee program...

Expected Result: 15 points for Threes

100 points for Yahtzee Bonus

Actual Result: 15 points for Threes

100 points for Yahtzee Bonus

Results match -- test PASSES

Now running test 13d...

Dice values are: 5-5-5-5-5

Player selects Fives

Sending input to Yahtzee program...

Expected Result: 25 points for Fives

100 points for Yahtzee Bonus

Actual Result: 25 points for Fives

100 points for Yahtzee Bonus

Results match -- test PASSES

Now running test 14d...

Requesting Upper Scoring from Yahtzee program...

Expected Result:

Upper Section Subtotal: 90

Bonus: 35

Total of Upper Section: 125

Actual Result:

Upper Section Subtotal: 90

Bonus: 35

Total of Upper Section: 125

Results match -- test PASSES

Now running test 15d...

Requesting Lower Scoring from Yahtzee program...

Expected Result:

Yahtzee Bonus: 700

Total of Lower Section: 850

Total of Upper Section: 125

Grand Total: 975

Actual Result:

Yahtzee Bonus: 700

Total of Lower Section: 850

Total of Upper Section: 125

Grand Total: 975

Results match -- test PASSES

BUILD SUCCESSFUL (total time: 1 second)

**Some Thoughts on Project 3**

The Test Plan section of this report includes discussion as to how I chose my test cases and what they were meant to test, so I won’t cover that here. As far as assumptions go, I will note that it is very helpful (in keeping this project manageable) to assume that the Yahtzee Game won’t be receiving erroneous input. I feel that the scope of any test plan will depend on whether other teams might be testing other aspects of a project. It is easy to imagine that other components of a Yahtzee game which used this scoring program would need to be tested themselves to ensure correct inputs to the scoring program. There is always going to be a great deal of context involved when developing a test plan. The kinds of errors that might be possible when two units are integrated will be very different than the kinds of errors that might be possible with units that directly interface with a human user. When you are integrating two units, there will be a sense where you must choose with which unit the buck truly stops (which team has to actually meaningfully deal with a potential error).

I thought that our study of boundary values and equivalence classes was helpful when thinking about test cases. There are way too many ways to get scores in the various Yahtzee categories to test them all. Having a way of thinking about these categories as having boundaries is important.

Revisiting the idea of test-driven development was interesting here. I found TDD to be both useful and constraining. Contending with a relatively small number of test cases for each programming effort helped in not getting overwhelmed. TDD also helps to suggest a starting point when writing code. There were times, however, when it was clear to me that a certain method should be reusable (like the method that checks by count whether a 3-of-a-Kind, 4-of-a-Kind, Full House, or Yahtzee has been hit). In those cases, it was just too tempting to make sure the initial design of a method could be easily scaled to do everything I would later want it to do.

The requirement for Full House demonstrated how important it is to properly understand the requirements for a project. This just shows that it will be important to talk to the customer (or the boss) frequently to make sure effort isn’t wasted going down the wrong path. I also found it interesting that I’d been playing Yahtzee wrong (with regard to Yahtzee Bonuses) my whole life. Other students I talked to had been playing wrong as well, so there is a lesson about not making unwarranted assumptions. I also found myself revising many different aspects of this project as I went along. I couldn’t correctly anticipate the exact test cases I’d ultimately want in some cases until I had made more progress. I had to think about how the cases would interact with each other and be grouped with each other. This experience makes me appreciate how challenging it would be to develop and test a large piece of software.

**Code for the Test Harness Software**

package ceg3110project3;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

/\*\*

\* @author Daniel -- Test harness for YahtzeeGame class.

\*/

public class CEG3110Project3 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws FileNotFoundException {

// indices to look at specific input in each test case

final int TEST\_NUMBER = 0;

final int CATEGORY = 1;

final int DICE = 2;

final int SCORE = 3;

final int BONUS = 4;

char gameNum = 'A';

String[] testCase;

String actualResult, expectedResult;

File inputFile = new File("inputBank.txt");

Scanner input = new Scanner(inputFile);

// continue reading input file while there is still data

while (input.hasNext()) {

YahtzeeGame game = new YahtzeeGame();

System.out.println("---------------------------------------------");

System.out.println("Game " + gameNum + ":"); // print game number

// eat and ignore the blank line between games in the input file

if (gameNum > 'A') {

input.nextLine();

}

// run the test cases for individual category scoring

for (int i = 0; i < 13; i++) {

// each line is a comma separated test case

testCase = input.nextLine().split(",");

// trim whitespace from each element of test case

for (int j = 0; j < testCase.length; j++) {

testCase[j] = testCase[j].trim();

}

// display info on test number, dice rolled, and category chosen

System.out.println("Now running test " + testCase[TEST\_NUMBER]

+ "...");

System.out.println("Dice values are: " + testCase[DICE]);

System.out.println("Player selects " + testCase[CATEGORY]);

System.out.println("\nSending input to Yahtzee program...\n");

expectedResult = testCase[SCORE];

// if expecting a Yahtzee Bonus, include in expected result

if (testCase.length > BONUS) {

expectedResult += "\n\t\t " + testCase[BONUS];

}

System.out.println("Expected Result: " + expectedResult);

// get the actual result from YahtzeeGame and display

actualResult

= game.processScoreRequest(testCase[DICE], testCase[CATEGORY]);

System.out.println(" Actual Result: " + actualResult + "\n");

// if the expected result matches actual result, test passes

if (expectedResult.equals(actualResult)) {

System.out.println("Results match -- test PASSES\n\n");

} else { // if not, it fails

System.out.println("Results differ -- test FAILS\n\n");

}

}

// we now read test case 14, for Upper Scoring

testCase = input.nextLine().split(",");

for (int j = 0; j < testCase.length; j++) {

testCase[j] = testCase[j].trim();

}

System.out.println("Now running test " + testCase[TEST\_NUMBER]

+ "...");

System.out.println("\nRequesting Upper Scoring from Yahtzee "

+ "program...\n");

// grab expected Upper Scoring from test case

expectedResult = "";

for (int j = 1; j < testCase.length; j++) {

expectedResult += "\n\t" + testCase[j];

}

// get the Upper Scoring result from YahtzeeGame

actualResult = game.getUpperScoring();

System.out.println("Expected Result:" + expectedResult);

System.out.println("Actual Result:" + actualResult);

// if expected Upper Scoring matches actual Upper Scoring, passes test

if (expectedResult.equals(actualResult)) {

System.out.println("\nResults match -- test PASSES\n\n");

} else { // otherwise fails test

System.out.println("\nResults differ -- test FAILS\n\n");

}

// we now read test case 15, for Lower Scoring

testCase = input.nextLine().split(",");

for (int j = 0; j < testCase.length; j++) {

testCase[j] = testCase[j].trim();

}

System.out.println("Now running test " + testCase[TEST\_NUMBER]

+ "...");

System.out.println("\nRequesting Lower Scoring from Yahtzee "

+ "program...\n");

// grab expected Lower Scoring from test case

expectedResult = "";

for (int j = 1; j < testCase.length; j++) {

expectedResult += "\n\t" + testCase[j];

}

// get the Lower Scoring result from YahtzeeGame

actualResult = game.getLowerScoring();

System.out.println("Expected Result:" + expectedResult);

System.out.println("Actual Result:" + actualResult);

// if expected Lower Scoring matches actual Upper Scoring, passes test

if (expectedResult.equals(actualResult)) {

System.out.println("\nResults match -- test PASSES\n\n");

} else { // otherwise fails test

System.out.println("\nResults differ -- test FAILS\n\n");

}

// increment game number in case input file has multiple games

gameNum++;

}

}

}

**Code for the Yahtzee Software**

/\*

\* Daniel Nedrow

\* CEG 3110

\* Project 3

\* Professor John Reisner

\*/

package ceg3110project3;

/\*\*

\* @author Daniel This class calculates individual category scores in both

\* sections of a Yahtzee score sheet, and also calculates the upper section

\* bonus and totals for all sections.

\*/

public class YahtzeeGame {

final String[] CATEGORIES = {"Aces", "Twos", "Threes", "Fours", "Fives",

"Sixes", "3-of-a-Kind", "4-of-a-Kind", "Full House", "Small Straight",

"Large Straight", "Yahtzee", "Chance"};

final int YAHTZEE\_INDEX = 11;

final int NUM\_CATEGORIES = 13;

int[] individualCategoryScores;

int subTotalUpper, bonusUpper, totalUpper, numPlays, yahtzeeBonus,

totalLower, grandTotal;

boolean yahtzeePreviouslyRolled;

/\*\*

\* Creates a new single-player Yahtzee game.

\*/

public YahtzeeGame() {

individualCategoryScores = new int[NUM\_CATEGORIES];

}

/\*\*

\* Takes a string representing the dice rolled and a string representing the

\* players category choice, and calls appropriate method to find score. The

\* method will then return a String which gives the score for given dice

\* roll in chosen category.

\*

\* @param diceString the 5 dice the player rolled

\* @param choice the category the player wishes to score

\* @return a String which gives the category score for the given dice roll

\*/

public String processScoreRequest(String diceString, String choice) {

String result;

// get the dice into an int array

String[] diceFaces = diceString.split("-");

int[] dice = new int[5];

for (int i = 0; i < dice.length; i++) {

dice[i] = Integer.parseInt(diceFaces[i]);

}

// get an index of which category was chosen

int index;

for (index = 0; index < CATEGORIES.length; index++) {

if (CATEGORIES[index].equals(choice)) {

break;

}

}

// get an individual score from either upper or lower section category

if (index < 6) {

result = getUpperCategoryScore(dice, index);

} else {

result = getLowerCategoryScore(dice, index);

}

// check Yahtzee Bonus if current category was NOT Yahtzee AND Yahtzee

// had already been rolled

if (index != YAHTZEE\_INDEX && yahtzeePreviouslyRolled

&& checkByCount(dice, YAHTZEE\_INDEX)) {

yahtzeeBonus += 100;

result += ("\n\t\t 100 points for Yahtzee Bonus");

}

return result;

}

/\*\*

\* Calculate category score for any of the upper categories, and return a

\* String which gives the score for given dice roll in that category.

\*

\* @param dice the 5 dice the player rolled

\* @param index the index of the category chosen

\* @return a String which gives the category score for the given dice roll

\*/

public String getUpperCategoryScore(int[] dice, int index) {

int match = index + 1; // converts index into dice face we look for

// add all dice matching category chosen to that category's score

for (int die : dice) {

if (die == match) {

individualCategoryScores[index] += die;

}

}

return (individualCategoryScores[index]

+ " points for " + CATEGORIES[index]);

}

/\*\*

\* Calculate category score for any of the lower categories, and return a

\* String which gives the score for given dice roll in that category.

\*

\* @param dice the 5 dice the player rolled

\* @param index the index of the category chosen

\* @return a String which gives the category score for the given dice roll

\*/

public String getLowerCategoryScore(int[] dice, int index) {

// calculate category score based on category chosen

switch (index) {

case 6: // 3-of-a-Kind (scores all 5 dice like 4-of-a-Kind)

case 7: // 4-of-a-Kind

if (checkByCount(dice, index)) { // check for 3 or 4 die matches

individualCategoryScores[index] = sumDice(dice);

} else {

individualCategoryScores[index] = 0;

}

break;

case 8: // Full House

if (checkByCount(dice, index)) { // check 3-of-a-Kind + 2-Pair

individualCategoryScores[index] = 25;

} else {

individualCategoryScores[index] = 0;

}

break;

case 9: // Small Straight

if (checkStraights(dice, index)) { //check for Small Straight

individualCategoryScores[index] = 30;

} else {

individualCategoryScores[index] = 0;

}

break;

case 10: // Large Straight

if (checkStraights(dice, index)) { //check for Large Straight

individualCategoryScores[index] = 40;

} else {

individualCategoryScores[index] = 0;

}

break;

case 11: // Yahtzee

if (checkByCount(dice, index)) { // check for 5 die matches

individualCategoryScores[index] = 50;

yahtzeePreviouslyRolled = true; // now eligible for bonuses

} else {

individualCategoryScores[index] = 0;

}

break;

case 12: // Chance

individualCategoryScores[index] = sumDice(dice);

break;

}

return (individualCategoryScores[index]

+ " points for " + CATEGORIES[index]);

}

/\*\*

\* Check that a sufficient number of dice match each other for a given

\* category. Useful for 3-of-a-Kind, 4-of-a-Kind, Full House, and Yahtzee.

\*

\* @param dice the 5 dice the player rolled

\* @param index the index of the category chosen

\* @return true if the category requirements are satisfied, false otherwise

\*/

public boolean checkByCount(int[] dice, int index) {

int countNeeded; // number of matches required (based on category)

// use the category selected to determine the matches required

switch (index) {

case 7: // 4-of-a-Kind

countNeeded = 4;

break;

case 11: // Yahtzee

countNeeded = 5;

break;

default: // 3-of-a-Kind or Full House's primary requirement

countNeeded = 3;

break;

}

// get a count of 1's, 2's, 3's, 4's, 5's, and 6's

int[] counts = new int[6];

for (int i = 0; i < dice.length; i++) {

counts[dice[i] - 1]++;

}

// determine if category requirements were met or not

if (index != 8) { // 3-of-a-Kind, 4-of-a-Kind, Yahtzee

// if we find the number of matches we need (or more) return true

for (int count : counts) {

if (count >= countNeeded) {

return true;

}

}

} else { // for Full House check only

boolean foundCountOf3 = false, foundCountOf2 = false;

// check if we 3 dice hitting one value and 2 dice hitting another

for (int count : counts) {

if (count == countNeeded) {

foundCountOf3 = true;

}

if (count == 2) {

foundCountOf2 = true;

}

if (count == 5) {

return true; // a 5-of-a-Kind instantly counts as Full House

}

}

// return true if Full House requirements hit, false otherwise

return foundCountOf3 && foundCountOf2;

}

return false; // we missed the category requirements above

}

/\*\*

\* Check that a sufficient number of consecutive dice were rolled for either

\* a Small Straight (4 consecutive dice) or a Large Straight (5 consecutive

\* dice).

\*

\* @param dice the 5 dice the player rolled

\* @param index the index of the category chosen

\* @return true if the category requirements are satisfied, false otherwise

\*/

public boolean checkStraights(int[] dice, int index) {

// define 4 or 5 consecutive dice needed based on straight type

int consecutiveNumsNeeded = ((index == 9) ? 4 : 5);

int remainingNumsNeeded = consecutiveNumsNeeded; // for tracking

// get a count of 1's, 2's, 3's, 4's, 5's, and 6's

int[] counts = new int[6];

for (int i = 0; i < dice.length; i++) {

counts[dice[i] - 1]++;

}

// Check if player has number of consecutive dice needed for straight

for (int count : counts) {

if (count >= 1) { // decrement num needed if consecutive dice found

remainingNumsNeeded--;

} else { // reset num needed if gap is found in consecutive dice

remainingNumsNeeded = consecutiveNumsNeeded;

}

// if we found the number of consecutive dice needed, return true

if (remainingNumsNeeded == 0) {

return true;

}

}

return false; // we didn't find the number of consecutive dice needed

}

/\*\*

\* Returns the sum of the 5 dice the player rolled.

\*

\* @param dice the five dice the player rolled

\* @return the sum of these 5 dice

\*/

public int sumDice(int[] dice) {

int sumDice = 0;

for (int die : dice) {

sumDice += die;

}

return sumDice;

}

/\*\*

\* Return the Upper Section Subtotal, Bonus, and Total of Upper Section as a

\* String.

\*

\* @return a String containing Upper Section Subtotal, Bonus, and Total of

\* Upper Section

\*/

public String getUpperScoring() {

// get the subtotal

for (int i = 0; i < 6; i++) {

subTotalUpper += individualCategoryScores[i];

}

// check the bonus

if (subTotalUpper >= 63) {

bonusUpper += 35;

}

totalUpper = subTotalUpper + bonusUpper;

return "\n\tUpper Section Subtotal: " + subTotalUpper + "\n\tBonus: "

+ bonusUpper + "\n\tTotal of Upper Section: " + totalUpper;

}

/\*\*

\* Return the Yahtzee Bonus, Total of Lower Section, Total of Upper Section,

\* and Grand Total as a String.

\*

\* @return a String containing the Yahtzee Bonus, Total of Lower Section,

\* Total of Upper Section, and Grant Total

\*/

public String getLowerScoring() {

// get the total of lower section

for (int i = 6; i < 13; i++) {

totalLower += individualCategoryScores[i];

}

totalLower += yahtzeeBonus; // add in the Yahtzee Bonus score

grandTotal = totalUpper + totalLower;

return "\n\tYahtzee Bonus: " + yahtzeeBonus + "\n\tTotal of Lower "

+ "Section: " + totalLower + "\n\tTotal of Upper Section: "

+ totalUpper + "\n\tGrand Total: " + grandTotal;

}

}