In vague terms, the system shall be able to take the dice rolls as inputs, and fill in the score card at a specific space with a properly calculated score. Additionally, logic needs to be included to add in any bonuses in situations where required, such as having 63 points on the top half of the score card. Furthermore, the system must not allow for a new score to be put into a space that is already occupied.

The specifics for the requirements have been lain out below in a case by case basis.

**Ones space:**

The system shall calculate the score to be 1 times the number of ones rolled, and input this total into the ones space on the score card.

Test Cases:

*Test scenario where zero ‘1’s are rolled*

Input: [ 5, 2, 2, 3, 4 ] , 1 category

Expected Output: 0 scored in the 1 category

Actual Output:

Pass/Fail:

*Test scenario where one ‘1’ is rolled*

Input: [ 1, 2, 2, 3, 4 ] , 1 category

Expected Output: 1 scored in the 1 category

Actual Output:

Pass/Fail:

*Test scenario where two ‘1’s are rolled*

Input: [ 1,3,4,1,6 ], 1 category

Expected Output: 2 scored in the 1 category

Actual Output:

Pass/Fail:

*Test scenario where three ‘1’s are rolled*

Input: [ 1,4,1,6,1], 1 category

Expected Output: 3 scored in the 1 category

Actual Output:

Pass/Fail:

*Test scenario where four ‘1’s are rolled*

Input: [1,1,1,1,4], 1 category

Expected Output: 4 scored in the 1 category

Actual Output:

Pass/Fail:

*Test scenario where all dice rolled are ‘1’*

Input: [1,1,1,1,1] , 1 category

Expected Output: 5 scored in the 1 category

Actual Output:

Pass/Fail:

**Twos space:**

The system shall calculate the score to be 2 times the number of twos rolled, and input this total into the twos space on the score card.

Test Cases:

*Test scenario where zero ‘2’s are rolled*

Input: [ 1, 1, 5, 3, 4 ] , 2’s category

Expected Output: 0 scored in the 2’s category

Actual Output:

Pass/Fail:

*Test scenario where one ‘2’ is rolled*

Input: [ 1, 2, 5, 3, 4 ] , 2’s category

Expected Output: 2 scored in the 2’s category

Actual Output:

Pass/Fail:

*Test scenario where two ‘2’s are rolled*

Input: [ 2,3,4,1,2], 2’s category

Expected Output: 4 scored in the 2’s category

Actual Output:

Pass/Fail:

*Test scenario where three ‘2’s are rolled*

Input: [ 2,4,2,6,2], 2’s category

Expected Output: 6 scored in the 2 category

Actual Output:

Pass/Fail:

*Test scenario where four ‘2’s are rolled*

Input: [2,2,2,1,2], 2 category

Expected Output: 8 scored in the 2’s category

Actual Output:

Pass/Fail:

*Test scenario where all dice rolled are ‘2’*

Input: [2,2,2,2,2] , 2’s category

Expected Output: 10 scored in the 2’s category

Actual Output:

Pass/Fail:

**Three’s space:**

The system shall calculate the score to be 3 times the number of threes rolled, and input this total into the threes space on the score card.

Test Cases:

*Test scenario where zero ‘3’s are rolled*

Input: [ 1, 1, 5, 6, 4 ] , 3s category

Expected Output: 0 scored in the 3’s category

Actual Output:

Pass/Fail:

*Test scenario where one ‘3’ is rolled*

Input: [ 1, 6, 2, 4, 3], 3 category

Expected Output: 3 scored in the 3 category

Actual Output:

Pass/Fail:

*Test scenario where two ‘3’s are rolled*

Input: [ 1,3,4,1,3 ], 3 category

Expected Output: 6 scored in the 3 category

Actual Output:

Pass/Fail:

*Test scenario where three ‘3’s are rolled*

Input: [ 1,3,3,6,3], 3 category

Expected Output: 9 scored in the 3 category

Actual Output:

Pass/Fail:

*Test scenario where four ‘3’s are rolled*

Input: [3,3,1,3,3], 3 category

Expected Output: 12 scored in the 3 category

Actual Output:

Pass/Fail:

*Test scenario where all dice rolled are ‘3’*

Input:[3,3,3,3,3] , 3 category

Expected Output: 15 scored in the 3 category

Actual Output:

Pass/Fail:

**fours space:**

The system shall calculate the score to be 4 times the number of fours rolled, and input this total into the fours space on the score card.

Test Cases:

*Test scenario where zero ‘4’s are rolled*

Input: [ 1, 1, 5, 3, 5] , 4’s category

Expected Output: 0 scored in the 4’s category

Actual Output:

Pass/Fail:

*Test scenario where one ‘4’ is rolled*

Input: [ 1, 2, 2, 3, 4 ] , 4 category

Expected Output: 4 scored in the 4 category

Actual Output:

Pass/Fail:

*Test scenario where two ‘4’s are rolled*

Input: [ 1,3,4,1,4], 4 category

Expected Output: 8 scored in the 4 category

Actual Output:

Pass/Fail:

*Test scenario where three ‘4’s are rolled*

Input: [ 1,4,1,4,4], 4 category

Expected Output: 12 scored in the 4 category

Actual Output:

Pass/Fail:

*Test scenario where four ‘4’s are rolled*

Input: [4,4,4,1,4], 4 category

Expected Output: 16 scored in the 4 category

Actual Output:

Pass/Fail:

*Test scenario where all dice rolled are ‘4’*

Input: [4,4,4,4,4] , 4 category

Expected Output: 20 scored in the 4 category

Actual Output:

Pass/Fail:

**fives space:**

The system shall calculate the score to be 5 times the number of fives rolled, and input this total into the fives space on the score card.

Test Cases:

*Test scenario where zero ‘5’s are rolled*

Input: [ 1, 1, 2, 3, 4 ] , 5’s category

Expected Output: 0 scored in the 5’s category

Actual Output:

Pass/Fail:

*Test scenario where one ‘5’ is rolled*

Input: [ 5, 2, 2, 3, 4] , 5 category

Expected Output: 5 scored in the 5 category

Actual Output:

Pass/Fail:

*Test scenario where two ‘5’s are rolled*

Input: [ 1,3,5,1,5], 5 category

Expected Output: 10 scored in the 5 category

Actual Output:

Pass/Fail:

*Test scenario where three ‘5’s are rolled*

Input: [ 1,4,5,5,5], 5 category

Expected Output: 15 scored in the 5 category

Actual Output:

Pass/Fail:

*Test scenario where four ‘5’s are rolled*

Input: [5,5,5,5,4], 5 category

Expected Output: 20 scored in the 5 category

Actual Output:

Pass/Fail:

*Test scenario where all dice rolled are ‘5’*

Input: [5,5,5,5,5] , 5 category

Expected Output: 25 scored in the 5 category

Actual Output:

Pass/Fail:

**six space:**

The system shall calculate the score to be 6 times the number of sixes rolled, and input this total into the sixes space on the score card.

Test Cases:

*Test scenario where zero ‘6’s are rolled*

Input: [ 1, 1, 5, 3, 4 ] , 6’s category

Expected Output: 0 scored in the 6’s category

Actual Output:

Pass/Fail:

*Test scenario where one ‘6’ is rolled*

Input: [ 1, 2, 2, 3, 6] , 6 category

Expected Output: 6 scored in the 6 category

Actual Output:

Pass/Fail:

*Test scenario where two ‘6’s are rolled*

Input: [ 1,6,4,1,6 ], 6 category

Expected Output: 12 scored in the 6 category

Actual Output:

Pass/Fail:

*Test scenario where three ‘6’s are rolled*

Input: [ 1,6,6,6,1], 6 category

Expected Output: 18 scored in the 6 category

Actual Output:

Pass/Fail:

*Test scenario where four ‘6’s are rolled*

Input: [6,6,6,6,4], 6 category

Expected Output: 24 scored in the 6 category

Actual Output:

Pass/Fail:

*Test scenario where all dice rolled are ‘6’*

Input: [6,6,6,6,6] , 6 category

Expected Output: 30 scored in the 6 category

Actual Output:

Pass/Fail:

**Scores and bonuses:**

The system shall be able to properly apply bonuses that are accumulated by players for meeting certain conditions, and calculate the total score of the upper half of the card before and after this is applied.

*Test for proper sum of scores pre bonus:*

Input: [6,10,9,8,0,30]

Expected Output: 63

Actual Output:

Pass/Fail:

Input: [5,10,9,8,0,30]

Expected Output: 62

Actual Output:

Pass/Fail:

Input: [4,10,0,20,30]

Expected Output: 64

Actual Output:

Pass/Fail:

*Tests for acquisition condition of bonus: If total score is 63 or more, a bonus of 35 points is given.*

*Inputs are given in the format of [aces score, twos score, threes score, fives score, sixes score,]*

*Check boundary just on point of interest*

Input: [6,10,9,8,0,30]

Expected Output: 63, bonus added

Actual Output:

Pass/Fail:

*Check boundary just below point of interest*

Input: [5,10,9,8,0,30]

Expected Output: 62, bonus not added

Actual Output:

Pass/Fail:

*Check boundary just above point of interest*

Input: [4,10,0,20,30]

Expected Output: 64, bonus added

Actual Output:

Pass/Fail:

*Test that bonus is correctly added to scores where applicable*

*No invalid scores have been given as examples, because it is assumed that the method that checks if the bonus needs to be applied will stop this from being an issue. Scores lower than 63 should be flagged as not needing a bonus applied to them before this part of the program is reached. Similarly, the score totals have been given as raw numbers, since the method totaling the scores is being tested separately as well.*

*Input format (score , bonus)*

Input: 65 , 35

Expected Output: 100

Actual Output:

Pass/Fail:

Input: 67, 35

Expected Output: 102

Actual Output:

Pass/Fail:

*Test to ensure that if there is already a score in a space, the system will not allow a new score to replace it*

Precondition: There is already a score in the 1 spot

Input: [1,1,1,1,1], 1s place

Expected Output: Error: Score already present in designated space

Actual Output:

Pass/Fail:

Precondition: There is already a score in the 2 spot

Input: [1,1,1,1,1], 2s place

Expected Output: Error: Score already present in designated space

Actual Output:

Pass/Fail:

Precondition: There is already a score in the 3 spot

Input: [1,1,1,1,1], 3s place

Expected Output: Error: Score already present in designated space

Actual Output:

Pass/Fail:

Precondition: There is already a score in the 4 spot

Input: [1,1,1,1,1], 4s place

Expected Output: Error: Score already present in designated space

Actual Output:

Pass/Fail:

Precondition: There is already a score in the 5 spot

Input: [1,1,1,1,1], 5s place

Expected Output: Error: Score already present in designated space

Actual Output:

Pass/Fail:

Precondition: There is already a score in the 6 spot

Input: [1,1,1,1,1], 6s place

Expected Output: Error: Score already present in designated space

Actual Output:

Pass/Fail: