Get Lucky with Scheme

Lee county is organizing a county fair over the upcoming summer. You along with one other friend plan to participate in several games including the bucket ball game. Each player will be assigned a bucket to start with and the player that gets the bucket with the highest worth wins the game. The worth of each bucket is calculated as per the rules below.

A bucket contains 10 balls of four different colors – red, blue, white, and green. Each red is worth **5** points, each green is worth **4** points, each blue ball is worth **3** points, and each white ball is worth **1** point. You are going to write a computer program in *Scheme* to look at a bucket and decide how many points it's worth.

IMPLEMENTATION

In our program, we'll represent a ball by a word like R (red ball) or B (blue ball). A bucket will be a sentence of balls, like this:

(R B G R R R B W W R)

This bucket is worth 37 points: red (5), plus green (1), plus blue (2), plus white (2).

We have a particular program structure in mind. We'll describe all of the procedures you need to write; if you turn each description into a working procedure, then you should have a complete program. Our descriptions are ordered *bottom-up*, which means that for each procedure you will already have written the helper procedures you need. (This ordering will help you write the project, but it means that we're beginning with small details. If we were describing a project to help you understand its structure, we'd do it in *top-down* order, starting with the most general procedures.)

Ball-val

Write a procedure ball-val that takes a single ball as its argument and returns the value of that ball.

> (ball-val 'R)

> (ball-val 'G)

4

Count-balls

Write a procedure count-balls that takes a color and a bucket as arguments and returns the number of balls in the bucket with the given color.

> (count-balls 'R '(R B G R R R B W R W))

5

> (count-balls 'W '(W R R R R G B B G W))

2

Color-counts

Write a procedure color-counts that takes a bucket as its argument and returns a sentence containing the number of reds, the number of green, the number of blues, and the number of whites in the bucket.

> (color-counts '(R B G R R R B W R W))

'(5 1 2 2)

> (color-counts '(W R R R R G B B G W))

'(4 2 2 2)

Bucket-val

Write a procedure bucket-val that takes a bucket as its argument and returns the total number of points that the bucket is worth.

> (bucket-val '(R B G R R R B W R W))

37

> (bucket-val '(W R R R R G B B G W))

36

Judge

Write a procedure judge that takes two arguments bucket_1 and bucket_2 and returns who won the game player of BUCKET_1 or BUCKET_2.

> (judge '(R B G R R R B W R W) '(W R R R R G B B G W))

Bucket 1, Won .. !

> (judge '(R B G R G W B B R W) '(R B R G R W B B G W))

It's a Tie .. !

OUTPUT

Upon executing your Scheme code, we should be able to see the following items printed on a console:

(1) Judge Decision

(2) Bucket Value

(3) Color Counts, for both players

Sample Output 1:	**Sample Output 2:**
; Bucket_1 > (bucket-val '(R B G R R R B W R W)) 29 > (color-counts '(R B G R R R B W R W)) '(5 1 2 2)	; Bucket_2 > (bucket-val '(W R R R R G B B G W)) 28 > (color-counts '(W R R R R G B B G W)) '(4 2 2 2)

EXTRA CREDIT

If there are five or more identical balls in a given bucket of 10 balls, add 5 bonus points to the overall worth of a bucket. For instance,

> (bucket-val '(R B G R R R B W R W))

37 + 5 = 42

This bucket will be rewarded 5 extra points, because it has 5 red balls.