Name: _____ **Date:** _____

Testing the Edges Activity – Test Case Challenge Worksheet – Solutions

Instructions

Create a normal test case, an edge case, and an error case for each of the following function descriptions. In each case describe how you applied the design process in arriving at your solution.

1. The `sqrt`(double x) function (which computes the square root of its input value x).

   For the `sqrt` function:

   - **Normal cases: `sqrt(16)` = 4, `sqrt(5)` = approx. 2.236**
   - **Edge case: `sqrt(0)` = 0**
   - **Error case: `sqrt(-1)` (causes an error)**

   I applied the design process to analyze the problem of how to test the square root function. I had to decide what cases were relevant and then designed appropriate cases to test potential problem. To continue with the design process, these test cases would be implemented to see if the program functions correctly.

2. A `getBrightestComponent(int r, int g, int b)` function. This function takes in the red, green, and blue components of a color (each of which must range from 0 to 255 inclusive), and returns the brightest (highest value) among the three inputs.

   For the `getBrightestComponent` function:

   - **Normal cases:**
     - **`getBrightestComponent(26, 100, 50)` = 100**
     - **`getBrightestComponent(255, 128, 0)` = 255**
     - **`getBrightestComponent(36, 72, 108)` = 108**
   - **Edge cases:**
     - **`getBrightestComponent(0, 0, 0)` = 0**
     - **`getBrightestComponent(255, 255, 128)` = 255**
     - **`getBrightestComponent(0, 0, 1)` = 1**
   - **Error cases:**
     - **`getBrightestComponent(0, 0, -1)` = error**
     - **getBrightestComponent(1000, 2000, 3000) = error**

   The design process implementation for this function was similar to the square root above. First, there was analysis as to what the normal, edge and error cases would look like. I then designed the test cases and tested them to learn if they functioned correctly.