Name: Mubashar Khan **Assignment #:** 4

Summary of Problem Statement **Problem #** 3

Add data validation and controls to the program Asks the user to enter either a color band or a resistance and uses that information to find either the resistance or the color band.

Known / Input

q = 0; Defining the Counter
redo = 1; defines redo so the program begins the loop

Unknown / Output

none

Assumptions

none

Other Variables

k = 1; Defines counter
Resist = gets user input for the resistance
[row, col] = used to end the loop early if the user enters a valid resistance.
Multiplier = Resist(:,3:col); Redefines Multiplier to fit the size of the new matrix
k = k + 1; redefines counter

Algorithm

First the program needed to check the user input to make sure that it was valid. If the user entered a resistance that had a improper multiplier, the program prompts them to rewrite the resistance (up to 4 times)
If the user doesn't give a valid resistance, the program will take the most recent entered value and convert all values after the first 2 elements into zeros by taking the vector Multiplication and using .* 0 to make all values zero. This could also be accomplished using the zero() command but multiplying by 0 takes less code.

The second validation was done using a similar condition to the first. Using the condition, the program prompts the user reenter the colors if they enter too few or too many and if the user enter too few on the fourth try, the program is terminated. Additionally if the user enter more than 3 colors, the program only uses the last 3 colors using Color = {Color{(col-2)},Color{(col-1)},Color{(col)}}; where col is found using size(color).

The final control structure allows the program to repeat as many times as the user desires and once the user decides to stop the program, it displays the number of times the program was run.

Test Cases

Using the Test Case I created of {'blue','black','orange','green'} I tested the program.
Output: 300,000
This matches the output for {'black','orange','green'}