Q1: How many unique colour values can the colour value integer contain?

A: **2564 = 4,294,967,296**

Q2: What is the minimum value, maximum value, and range for each colour component?

A: **All components have a minimum of 0, a maximum of 255, and a range of 256.**

Q3: Suppose the red component of the RGBA colour is to be stored in an 8-bit integer (char) variable, and is set to the decimal value “unsigned char red = 94”. Write this value as a binary number.

A: **1011110**

Q4: The byte containing the red value (94) from question 3 is now to be stored in the RGBA colour value (in the left-most byte). Assuming all other colour bytes are initialized to 0, write the value of the 4-byte colour value integer in binary.

A: **01011110 00000000 00000000 00000000**

Q5: What is the decimal value of the binary number from question 4?

A: **1,577,058,304**

Q6: Write the bit shifting operation (in C++) that will move all bits from the ‘R’ position in the colour variable to the ‘G’ position.

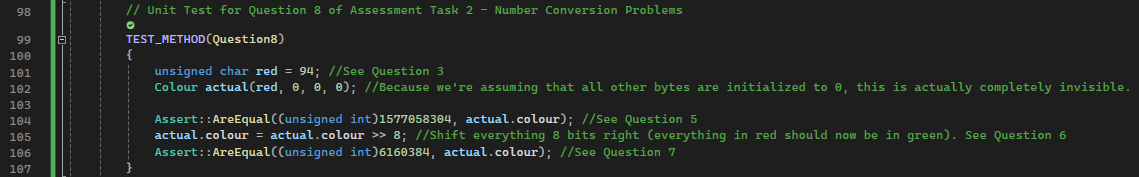
A: **colour = colour >> 8;**

Q7: Our colour value now has the green colour component set, and no red, blue, or alpha colour component values. What are the decimal and binary value of the colour variable now?

A (Binary): **00000000 01011110 00000000 00000000**

A (Decimal): **6,160,384**

Q8: After you have created your Colour class and implemented all the functions listed in the class definition above, add at least 1 new unit test to the unit test program using your answers in this exercise to verify your code.

****A: **See ColourTests.cpp (line 99)**