### Number Conversion Exercises

Along with the Vector and Matrix classes, your math library will contain a class that encapsulates an RGBA (red, green, blue, alpha) colour, stored as a 4 byte integer where each colour component is stored in a single byte.

The Colour class defines the following variables and functions:

public class Colour

{

public UInt32 colour;

public Colour() {}

public Colour(byte red, byte green, byte blue, byte alpha) {}

public byte GetRed() {}

public void SetRed(byte red) {}

public byte GetGreen() {}

public void SetGreen(byte green) {}

public byte GetBlue() {}

public void SetBlue(byte blue) {}

public byte GetAlpha(){}

public void SetAlpha(byte alpha) {}

}

To guide you through the development and testing of this class, answer the following questions:

1. How many unique colour values can the *colour* variable contain?

256

1. What is the minimum value, maximum value, and range for each colour component?

0, 255, 255

1. 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   
     
   　 char red = 94  
     
   Write this value as a binary number

1011110

1. 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 variable in binary:

01011110 00000000 00000000 00000000

1. What is the decimal value of the binary number from question 4?

94 0 0 0

1. Write the bit shifting operation (in C#) that will move all bits from the ‘R’ position in the colour variable to the ‘G’ position.  
     
   Colour >> 8
2. 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?

6,160,384

00000000 01011110 00000000 00000000

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

[TestMethod]

public void ColourTestFunction()

{

Colour c = new Colour(0x94, 0x00, 0x00, 0x00);

c.SetGreen(c.GetRed());

Assert.AreEqual<byte>(.GetGreen(), 0x94);

}

**NOTE: submit your answers to these exercises with your assessment**