Problem 1: Gray Images

Gray-scale images are much easier to process than colour images. Pixels in colour images have red, green, and blue (RGB) components, each in range 0 to 255. Having to process three components make things more challenging. Instead, pixels in a gray-scale image have a single gray component, in the range 0 to 255, where 0 is black, 255 is white, and all the other values are shades of gray. To make image processing easier, we first need to write a program to translate a colour image into a gray-scale image.

The algorithm for this is simple, each pixel in a colour image is converted to a corresponding gray pixel using an average of the three colour components:

$$I = \frac{R + G + B}{3}$$

where R, G, and B are the value of the red, green, and blue components of a pixel, and I is the resulting shade of gray. Setting all three of the pixel's colour components to I transforms it into a gray pixel.

Write a program called `Problem1.java` that reads in a colour image file, transforms it to a gray-scale image and writes it to a new image file. The program should load the image using the provided `ImageRW` code (see Hints and Suggestions in the a2.pdf file), convert all pixels in the image to their corresponding gray intensity (using the formula above), and then write the resulting image to the specified output file (using the provided ImageRW) code.

Input

The names of the input and output image files, separated by a white-space.

Output

The program produces the specified image file and does not print anything to the console. You can view the output file using a standard image viewer on your computer.

Sample Input	**Sample Output**
`ball.png ball_gs.png`	*N/A*