Name and, if possible, ID#: [redacted]

Problem 2

Colors in Java can be represented by objects of type ***Color***. Each such object contains the ***red***, ***green*** and ***blue*** components of the corresponding color as integer values from **0** to **255**. Consider below a Java code that creates and initializes a rectangular array of ***Color*** type:

```
import java.util.Scanner;
import java.awt.Color;

public class Colors {

public static void main(String args[]) {
      Scanner in = new Scanner(System.in);

// Read number of rows and columns and create a Color array of such size
      Color[][] c = new Color[in.nextInt()][in.nextInt()];

// For each element read the red, green and blue components as integers and
// create a Color object by calling Color(int, int, int) constructor
      for (int row = 0; row < c.length; row++)
            for (int col = 0; col < c[0].length; col++)
                  c[row][col] = new Color(in.nextInt(), in.nextInt(), in.nextInt());

// TO BE CONTINUED
}
}
```

Continue with a Java code that creates another array ***Color[][] g*** of the same size and fills it with gray equivalents of the colors from the array ***Color[][] c***. To get a grey equivalent of a given color ***c[i][j]***, it is enough to construct a ***Color*** object, whose red, green and blue components all are equal to the calculated average of red, green and blue components of the initial ***c[i][j]***. Use ***int getRed()***, ***int getGreen()*** and ***int getBlue()*** methods of class ***Color***.

4

Color [ ] [ ] g = new Color [c. length] [c. length] ;

for (int row1 = 0 , row1 < g. length ; row1 ++)

for (int col1 = 0 ; col1 < g. [0]. length ; col1 ++) g

~~g [row1] [col1] = new Color (get Red( ) + get Green( ) + get Blue( )~~ ;

~~g [row1] [col1]~~ = new Color [~~int get Red ( ), int get Green,~~ ~~int get Blue~~] ; 3

~~g [row1] [col1] = new Color (~~

for (int ~~row1 = 0 ; row1 < g . length ; row1 ++~~)

for (int ~~col1 = 0 ; col1 < g . [0] . length ; col1 ++~~)

g [~~row1~~] [col1] = ~~new Color~~ [

g [row 1] [col 1] = new Color [~~k~~ K , K , K]

K = get Red ( ) + get ~~Blue~~ Blue ( ) + get Green ( ) ;

3

;