```
public class Charge
-----------------------------------------
       Charge(double x0, double y0, double q0) // location and charge
double potentialAt(double x, double y)         // potential @ (x,y) due to charge
String toString()                              // string representation
```

Write a program that takes a double value w from the command line, creates four Charge objects that are each distance w from the center of the screen (0.5, 0.5), and prints the potential at location (0.25, 0.5) due to the combined four charges. All four charges should have the same unit charge.

```
01 public class FourChargeClient
02 {
03    public static void main(String [] args)
04    {
05        // read in distance w from command line
06        double w = Double.parseDouble(args[0]);
07        // set up center of screen location
08        double cx = 0.5;
09        double cy = 0.5;
10
11        // Construct four charges
12        Charge c1 = new Charge(cx + w, cy, 1.0);    // East
13        Charge c2 =                                 // South
14        Charge c3 =                                 // West
15        Charge c4 =                                 // North
16
17        // Compute potentials at (.25, .5)
18        double px = 0.25;
19        double py = 0.5;
20        double v1 = c1.potentialAt(             );
21        double v2 =
22        double v3 =
23        double v4 =
24
25        // Output total potential
26        double sum =
27        System.out.println("Potential = " + sum);
28    }
29 }
```

```
public class Picture
---------------------------------------------------------------------
       Picture(String filename)   // create a picture from a file
       Picture(int w, int h)      // create a blank w-by-h picture
   int width()                    // return the width of the picture
   int height()                   // return the height of the picture
 Color get(int i, int j)          // return the color of pixel (i,j)
  void set(int i, int j, Color c) // set the color of pixel (i,j) to c
  void show()                     // display the image in a window
  void save(String filename)      // save the image to a file
```

Write a program that takes the name of a picture file as a command-line input, and creates three images, one that contains only the red components, one for only the green, and one for only the blue.

```
01 import java.awt.Color;
02 public class ColorSeparation
03 {
04    public static void main(String [] args)
05    {
06        // read in the picture specified on the command-line argument
07        Picture pic = new Picture(args[0]);
08        int width = pic.width();
09        int height = pic.height();
10
11        // create three empty pictures of the same dimension
12        Picture R = new Picture(width, height);
13        Picture G =
14        Picture B =
15
16        // separate colors
17        for (int x = 0; x < width; x++)
18        {
19            for (int y = 0; y < height; y++)
20            {
21                // color value of current pixel
22                Color c = pic.
23
24                int r = c.getRed();
25                int g =
26                int b =
27
28                R.set(x, y, new Color(r, 0, 0));
29                G
30                B
31            }
32        }
33        // display each one in its own window
34        R.show();
35
36
37    }
38 }
```

Given the API for the Charge and Picture class, fill out the following client program which visualizes the charge potential at every point in a unit square. The program should read in the charged particles from a file that looks like this:

```
9
.51 .63 -100
.50 .50   40
.50 .72   10
.33 .33    5
.20 .20  -10
.70 .70   10
.82 .72   20
.85 .23   30
.90 .12  -5
```

```
01 public class Potential
02 {
03    public static void main(String[] args)
04    {
05          int n =
06          Charge [] a =
07          for (int i = 0; i < n; i++)
08          {
09                double x =
10                double y =
11                double q =
12                a[i] =
13          }
14          final int SIZE = 512;
15          Picture pic = new Picture(             );
16          // Loop over all rows in the image
17          for (int row = 0; row < SIZE; row++)
18          {
19                // Loop over all columns in the image
20                for (int col                          )
21                {
22                      double v = 0.0;
23                      // Loop over all particles, summing the charge
24                      for (int i = 0; i < n; i++)
25                      {
26                            double x = 1.0 * row / SIZE;
27                            double y = 1.0 * col / SIZE;
28                            v += a[i].                    // Add potential at x, y
29                      }
30                      // Convert the potential to a pretty color
31                      v = 128 + v / 2.0e10;
32                      int t = 0;
33                      if      (v <   0) t = 0;
34                      else if (v > 255) t = 255;
35                      else              t = (int) v;
36                      Color c = Color.getHSBColor(t / 255.0f, .9f, .9f);
37                      pic.set(row, SIZE - 1 - col, c);  // Set the pixel to the color
38                }
39          }
40          pic.show();
41    }
42 }
```