Problem Set 2

Working with Standard Draw

Grading Standards (Homework Category)

Standards-based Grading (percentage of indicated max points):
Outstanding – 100%, Excellent – 95%, Acceptable – 87%, Unacceptable – 75%, No

1. I can produce an externally correct solution to each of the following problems as demonstrated through test case pass rate. (5 pts per problem)
2. I can produce an internally correct solution through proper use of standard draw, commenting, and otherwise professional style. (10 pts)

Note: The extension is not graded – but working on extensions when you are done is factored into your monthly professionalism score.

Standard Draw Documentation Found At:

introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html

Problem Set 2

Problem 0 – Map it Up

Introduction

You are provided a list of longitude and latitude for cities in the Contiguous United States. Your task is to plot these cities as points to create a map. As a friendly reminder, I encourage you to make sure that you have the Standard Draw API nearby. Because the default canvas size in Standard Draw is 512 x 512 pixels, **you should resize the canvas** to avoid creating a square map of the United States. The dimensions of the canvas should be proportional to the ratio between the height and width of the coordinates in the input file. I.e., keep the canvas height at 512 pixels, but adjust the width accordingly.

Program Input

The first line of `usa.txt` contains information about the bounding box of the data. The first line of text contains four space-separated doubles which represent the minimum x and y-coordinates and then the maximum x- and y-coordinates for the entire text file. If you're really clever, you'll use these numbers to determine how to resize your canvas. The remaining lines contain two pieces of information on each line: a longitude and latitude, with the two being separated by three spaces.

Program Output

Your method, `america`, should create a plot of all the cities contained in `usa.txt`. No example is shown here because that would ruin the fun, wouldn't it? It would.

Problem Set 2

Problem 1 - Interwebs

Introduction

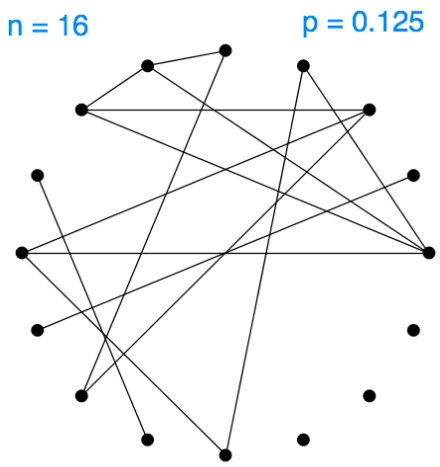
Write the method, `web`, that takes two parameters: an integer n and a double p (between 0 and 1). Then plot n equally spaced points along the circumference of a circle. Finally with probability p for each pair of points, draw a gray line connecting them.

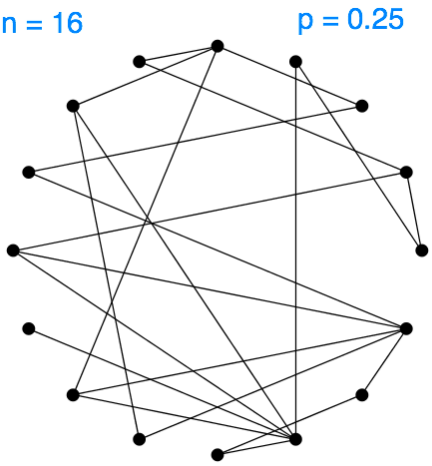
Program Input

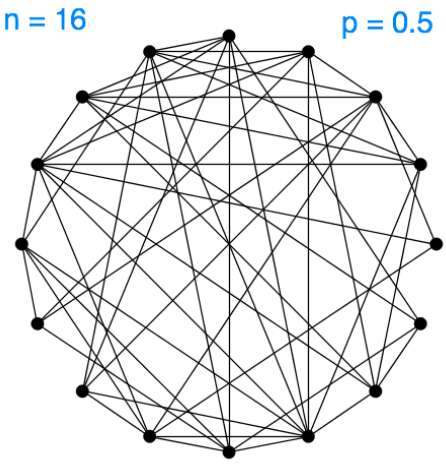
An `int` n represents the number of points to plot. A `double` p represents the probability that two points will be connected by a line.

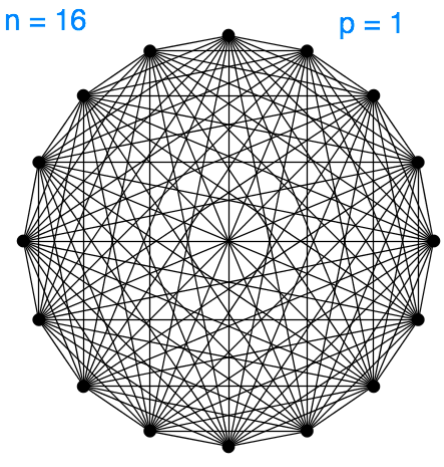
Program Output

Sample outputs are displayed below for a value of n `= 16` and varying values of p. Note that yours will not look identical due to the use of a randomness.









Problem Set 2

Problem 2 - Check Me Out, Yo

Introduction

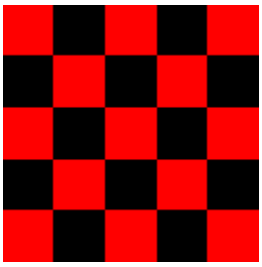
Write a method, `checkerboard`, that takes a single integer parameter, *N*, and plots an *N*-by-*N* checkerboard using a primary color and a secondary color of your choosing. As a friendly reminder, I encourage you to make sure that you have the Standard Draw API nearby. There you can explore what colors are part of the Standard Draw Library. You'll find filledSquare to be a helpful method. And if you prefer integers over decimals, you'll need to identify your coordinate system as well--see setScale.

Program Input

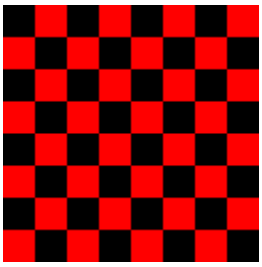
There is no text file to process for this problem. Simply write a method that takes a single integer parameter.

Program Output

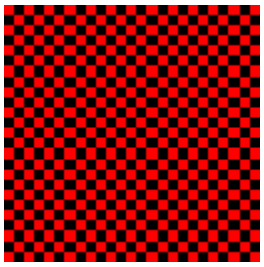
Below are sample checkerboards made of red and black with varying values of N. Note that the primary color red is always in the lower-left corner. Your checkerboard should fill the canvas.



N = 5



N = 8



N = 25

Problem Set 2

Problem 3 - Life is Rosy

Introduction

The polar coordinate system allows for a simpler, more elegant, and more intuitive way to model particular functions than the rectangular Cartesian system. For this problem you'll be plotting a function that generates a floral output. Write a method called `rose` that takes a single integer parameter, *n*, and plots a rose with *n* petals (if *n* is odd) or 2*n* petals (if *n* is even) by plotting the polar coordinates (r, θ) of the function $r = \sin(n\theta)$ for θ ranging from 0 to 2π radians.

Rather than generating a static image at the end, for this problem you should be creating an animation. An animation is nothing more than displaying a still image, pausing for a brief amount of time, and then displaying a new image. Think of a classic flip book for an example of an animation. You will need to read about the methods enableDoubleBuffering, show, and pause in the StandardDraw API.

Program Input

Take a single integer parameter, *n*.

Program Output

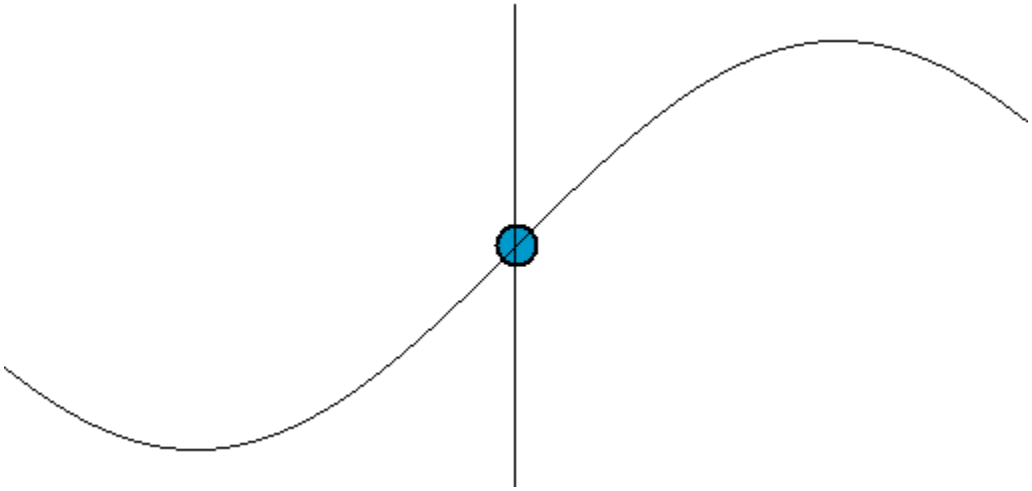
Below is the desired output for *n* = 4, 5, 8, and 9. Note that because this is a static piece of paper, you don't see the animation. The images below are of the final output once the animation has concluded.

Problem Set 2

Extension - Harmonic Motion

Introduction

Want to go beast mode? This problem is an extension for those who finish early. Write a method, `harmonicMotion`, that creates an animation (see enableDoubleBuffering, show, and pause) of simple harmonic motion, like the one seen below from Wikipedia.



Program Input

No input needed here. Just generate the animation when the method is invoked.

Program Output

Feel free to add some variation to the GIF above, but you should model simple harmonic motion.