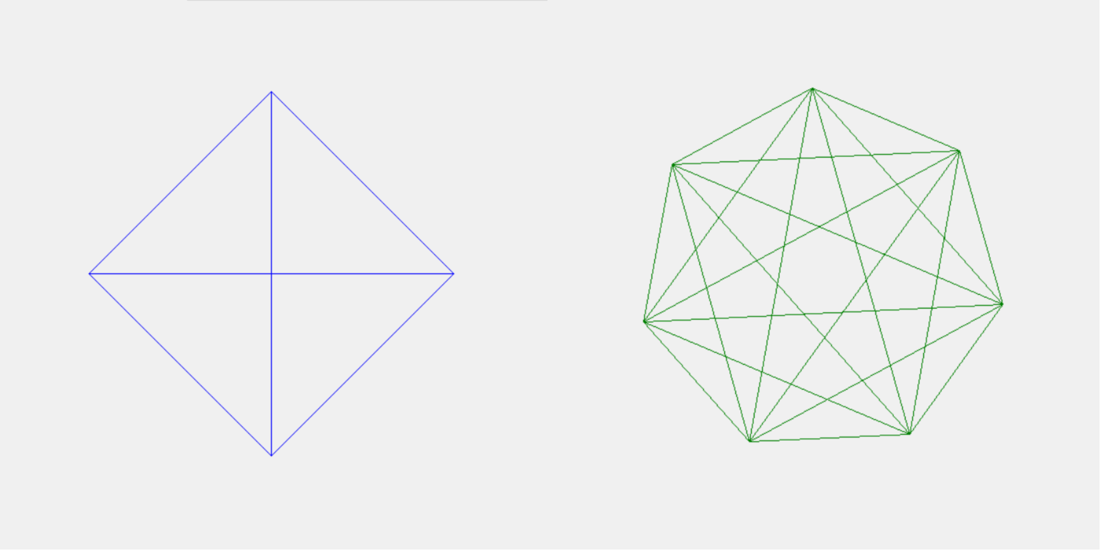
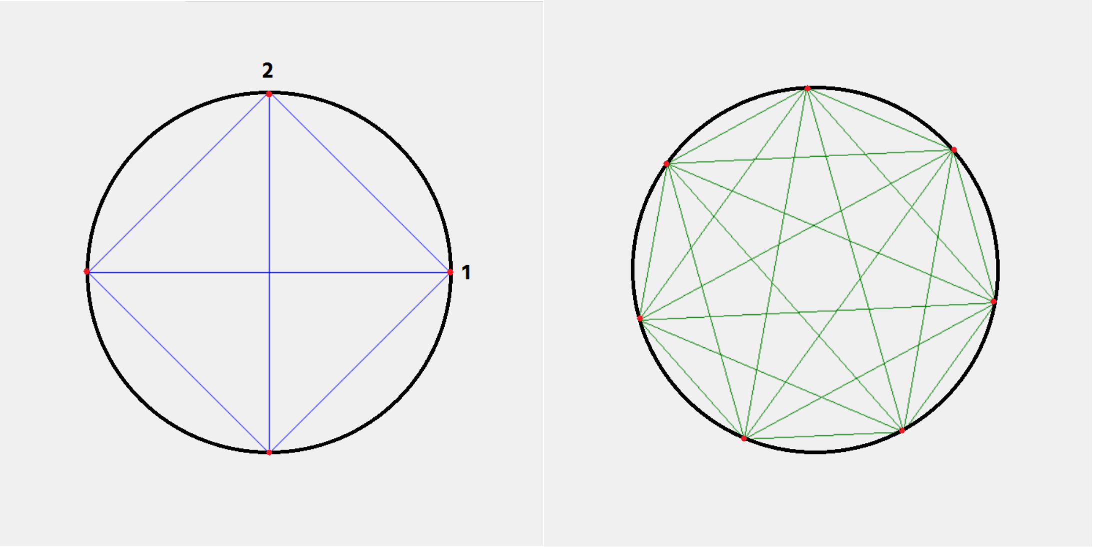
Homework 4.1

1. Fancy Wheel

These are two examples of a fancy wheel.



A fancy wheel is completely made out of lines. You can draw a fancy wheel by connecting each vertex to every other vertex. Each vertex is located on an “imaginary circle” around the wheel, as seen in the picture below.



You can determine the position of each vertex around the circle by calculating its angle from the horizontal, which increases by 2π/(# of vertices) from the previous vertex. For instance, vertex 2 on the 4-vertex fancy wheel differs by 2π/4 = 1/2π from vertex 1.

Keep in mind that the position of a point from the center (cx, cy) given the angle can be calculated by (x, y) = (cx + r cos θ, cy – r sin θ), where r is the radius.

Create an animated fancy wheel program with the following specifications:

- The window size is 600x600.
- The fancy wheel is positioned at the center of the window, and has a radius of 200.
- The fancy wheel has N vertices, where N is initially 4.
  - If the user presses the Right arrow or the Up arrow, N should increase by 1.
  - If the user presses the Left arrow or the Down arrow, N should decrease by 1. N must be at least 2—so it will be a line at minimum.
- The fancy wheel can be colored red, green, or blue. It is initially blue.
  - If the user presses r, the fancy wheel's color changes to red;
  - If the user presses g, the fancy wheel's color changes to green;
  - If the user presses b, the fancy wheel's color changes to blue.
- The fancy wheel must spin, either clockwise or anti-clockwise when the timer is fired. It initially spins clockwise. It turns for 10 degrees for every 100 milliseconds.
  - If the user clicks the mouse within the wheel (or "imaginary circle"), then the fancy wheel changes its spinning direction.
  - If the click falls outside the imaginary circle, the click is ignored.

We encourage you to approach this question one property at a time: work on generating a static 4-vertex blue fancy wheel at the correct location first; then, allow incrementing and decrementing number of vertices, followed by allowing color changes. Finally, add wheel rotation and mouse click controller.

Note: The top-level name of the function must be **runFancyWheel(),** taking no parameters. This is the function we will call to get the animation started.

Here is an example of an animated fancy wheel:
https://www.dropbox.com/s/5yzapvocs4jm8cb/Fancy%20Wheel.gif?dl=0