PROBLEM 5
HYPNO SPIRAL / SUNFLOWER
(LINK) / (LINK)

This p5.js sketch visually recreates the pattern seen in a sunflower head, known as a **phyllotaxis spiral**. At its core, the code simulates how seeds or florets are arranged in nature, relying on the mathematics of the golden angle to produce a naturally balanced distribution. The program begins by initializing a counter variable, `let n = 0;`, which keeps track of the number of circles that have been drawn. In the `setup()` function, a full-screen canvas is created with `createCanvas(windowWidth, windowHeight);`, ensuring that the artwork scales to the size of the browser window. The background is set to white with `background(255);`, and outlines are removed from the circles using `noStroke();`, so each point appears clean and distinct.

Within the `draw()` function, each animation frame calculates the location for a new circle, spreading outward from the canvas center. The center coordinates are stored in `cx` and `cy`. The key to the spiral pattern is how the angle for each step is determined. The code computes the angle using `let angle = n * radians(137.5);`. Here, 137.5 degrees is the so-called “golden angle”, which, when used in this way, ensures that every new circle is positioned so as not to overlap previous ones, imitating the efficient packing found in sunflower seeds. The distance from the center, or radius, grows with the square root of `n`, as shown by `let radius = 6 * sqrt(n);`. This scaling prevents the circles from bunching up as the spiral grows.

To find the exact position for each circle, the program converts from polar to Cartesian coordinates using the trigonometric functions: `let x = cx + radius * cos(angle);` and `let y = cy + radius * sin(angle);`. Color is also varied based on `n`, with the `fill()` function producing a smooth gradient effect as more circles are drawn: `fill(40 + n % 215, 160, 200, 180);`. Each new circle is drawn at its calculated position with `ellipse(x, y, 15, 15);`. After each circle is placed, the counter `n` increases by one, so the next frame continues the spiral. Finally, the code automatically stops when the spiral reaches the edge of the canvas—this is checked with `if (radius > min(width, height) / 2) noLoop();`, which halts the drawing process when further circles would exceed the canvas area.

Overall, the sketch elegantly combines geometry, mathematics, and color to imitate a sunflower’s spiral arrangement, all by carefully controlling position, angle, and appearance through the core functions and logic of p5.js.

```javascript
let n = 0; // Counter: number of steps (circles) drawn so far

function setup() {
  createCanvas(windowWidth, windowHeight); // Fullscreen canvas
  background(255);      // White background
  noStroke();           // Circles will have no outline
}

function draw() {
  let cx = width / 2;
  let cy = height / 2;

  // Calculate angle using golden angle for phyllotaxis spiral
  let angle = n * radians(137.5);
  // Radius increases like sqrt(n) to spread circles evenly
  let radius = 6 * sqrt(n);

  // Convert polar to cartesian coordinates
  let x = cx + radius * cos(angle);
  let y = cy + radius * sin(angle);

  // Assign a color based on n for a smooth gradient
  fill(40 + n % 215, 160, 200, 180);
  ellipse(x, y, 15, 15); // Draw one circle at (x, y)

  n++; // Next circle in next frame

  // Stop drawing when the spiral reaches the edge of the canvas
  if (radius > min(width, height) / 2) noLoop();
}

/*
  This sketch draws a static sunflower spiral, placing one colored circle at a time using the golden angle.
  The pattern grows outward from the center, with each circle fixed in place.
  Color and spacing are controlled by n; the process stops automatically at the canvas edge.
*/
```