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

The HYPNO SPIRAL code generates an animated spiral of circles, inspired by the Fibonacci pattern, using polar coordinates and smooth rotation for a hypnotic effect.

At the top, two global variables are defined: `t` (a global time variable for animation) and `rot` (which controls the rotation of the entire spiral). In `setup()`, the canvas is set to the full window with a black background, and shapes are drawn without fill, only outlines.

The main animation happens in `draw()`. Each frame, a semi-transparent black rectangle is drawn (`fill(10, 60); rect(...)`) to create a fading trail effect. The origin is moved to the center of the canvas with `translate()`. Then, the code uses `push()` and `pop()` to rotate the spiral by `rot` radians (`rotate(rot); drawSpiral();`). The variables `t` and `rot` are increased each frame, controlling both the pulsation and the spinning.

The function `drawSpiral()` contains the main loop, where for each point (`n`) up to `maxN`, the code computes the angle (using the golden angle in radians and a slight sinusoidal oscillation) and a radius that grows like `sqrt(n)`. These polar coordinates are converted to Cartesian (`x, y`), and each circle is drawn with a white outline whose thickness and diameter also oscillate in time, becoming smaller towards the outside.

The result is a clean, dynamic spiral made up of gently pulsing white circles, slowly rotating against a black background, for an elegant “hypno-spiral” visual effect.

The SUNFLOWER sketch also uses polar coordinates and the golden angle to create a spiral, as seen in the previous example.

Here, the global variable `n` counts the number of steps, and each frame in `draw()` adds one circle at a position computed from

`angle = n * radians(137.5)` and `radius = 6 * sqrt(n)`.

The `(x, y)` coordinates center each circle, and `fill(40 + n % 215, 160, 200, 180)` assigns a unique color to each.

Unlike the previous sketch, this one **does not animate or rotate** the spiral: every circle stays fixed, and `noLoop()` halts the drawing when the spiral reaches the canvas edge.

In summary, both codes use the same math, but this version builds a **static, step-by-step sunflower spiral** with changing colors and no motion.

```
let t = 0;    // Global time variable for animation
let rot = 0;  // Global rotation angle for the spiral

function setup() {
  createCanvas(windowWidth, windowHeight); // Fullscreen canvas
  background(10);     // Deep black background
  noFill();           // Draw only outlines (no fill)
  frameRate(60);      // Smooth animation
}

function draw() {
  // Fading effect: semi-transparent black rectangle for trails
  fill(10, 60);
  rect(0, 0, width, height);

  translate(width / 2, height / 2); // Move origin to center

  // Draw and rotate the spiral
  push();
  rotate(rot);    // Constant rotation for hypnotic effect
  drawSpiral();   // Draw all circles on the spiral
  pop();

  t += 0.017;     // Advance time for animation
  rot += 0.009;   // Increment rotation angle
}

function drawSpiral() {
  let maxN = 350;                    // Total number of circles
  let golden = radians(137.5);       // Golden angle in radians
```

```
  let maxN = 350;                    // Total number of circles
  let golden = radians(137.5);       // Golden angle in radians

  for (let n = 0; n < maxN; n++) {
    // Calculate spiral angle with small oscillation for "breathing" effect
    let angle = n * golden + 0.14 * sin(t + n * 0.08);
    // Radius grows like sqrt(n) for Fibonacci-style spiral
    let r = 12 + 18 * sqrt(n);

    // Convert polar to cartesian coordinates
    let x = r * cos(angle);
    let y = r * sin(angle);

    // Draw circle outline with animated thickness and size
    stroke(255, 180);
    strokeWeight(1.4 + 0.9 * sin(t * 0.7 + n * 0.22));
    let d = 15 + 5 * sin(t + n * 0.23) - n * 0.02; // Decreases outward
    ellipse(x, y, max(4, d), max(4, d));
  }
}

/*
  This code draws a hypnotic animated spiral of gently pulsing white circles.
  Each circle is placed using polar coordinates and the golden angle for a natural, Fibonacci-inspired pattern.
  The entire spiral rotates smoothly, creating an elegant, dynamic visual effect.
```