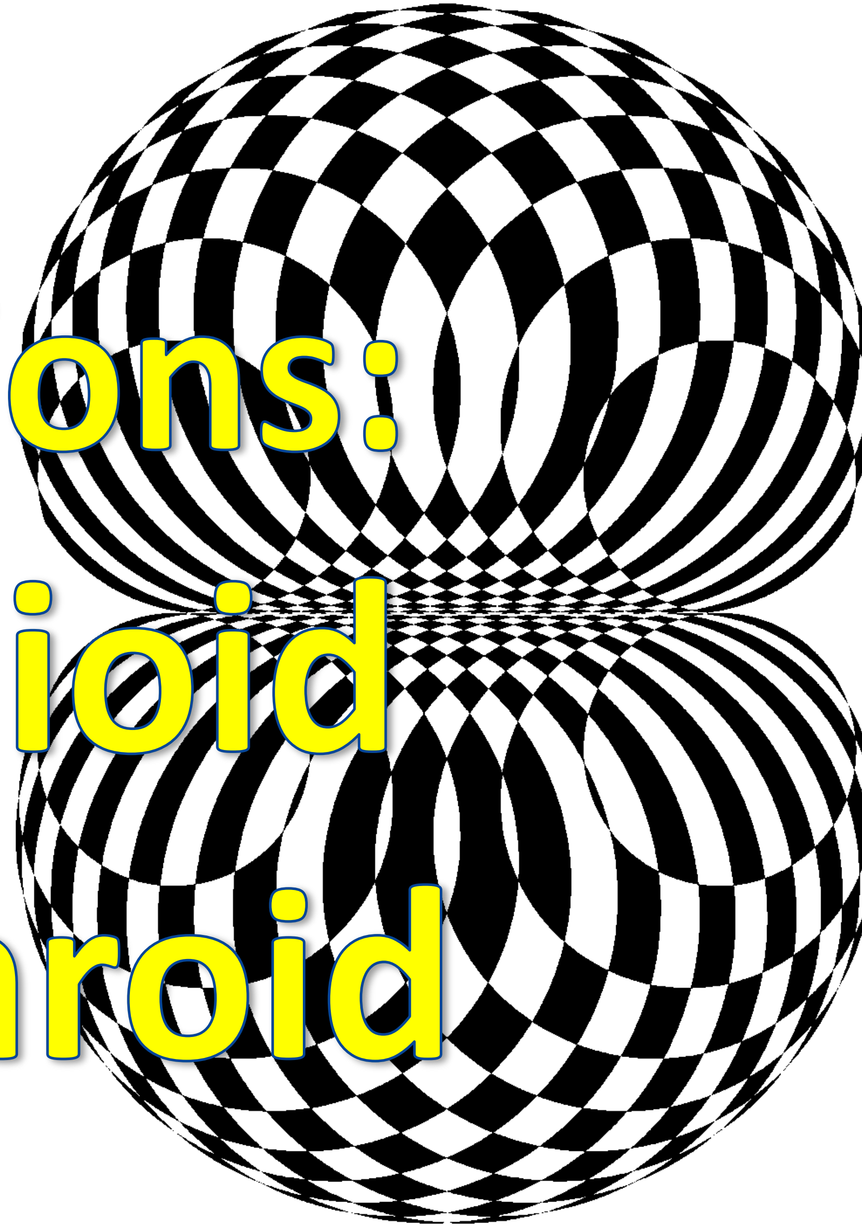
Circular constructions:

Cardioid
Nephroid
Ellipse



AF. January 2021

Cardioid

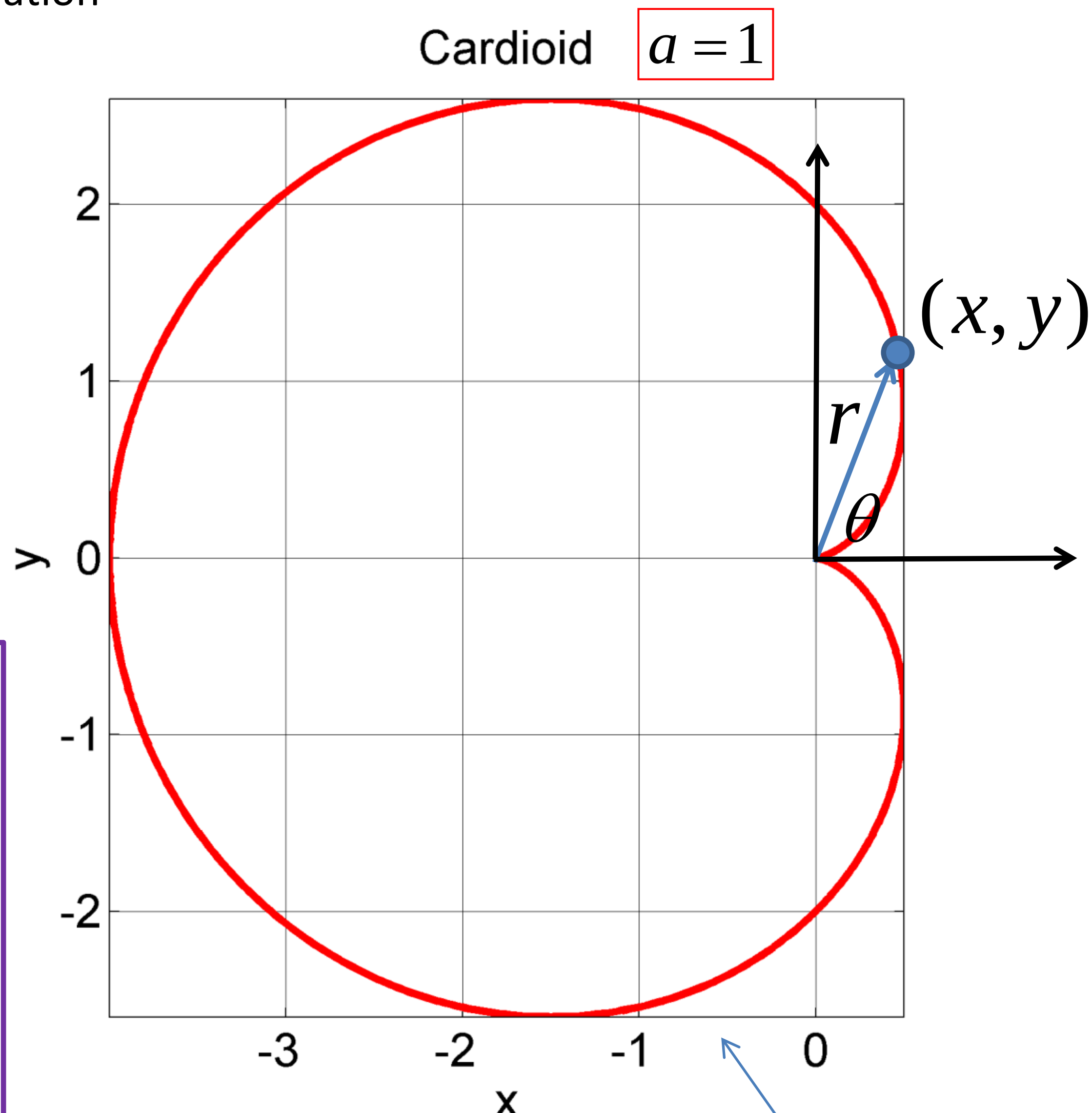
polar equation

$$r = 2a\left(1 - \cos\theta\right)$$

$$x = r\cos\theta$$

$$y = r\sin\theta$$

conversion to Cartesians



Coding challenge #1: Plot a cardioid

Define a *polar angle* θ vector of 2,000 elements, equally spaced between 0 and 2π radians.
Use the polar and Cartesian (x,y) equations above to plot a cardioid.
Output is a PNG graphics file.

https://en.wikipedia.org/wiki/Cardioid

Note you can easily achieve this in Excel too.

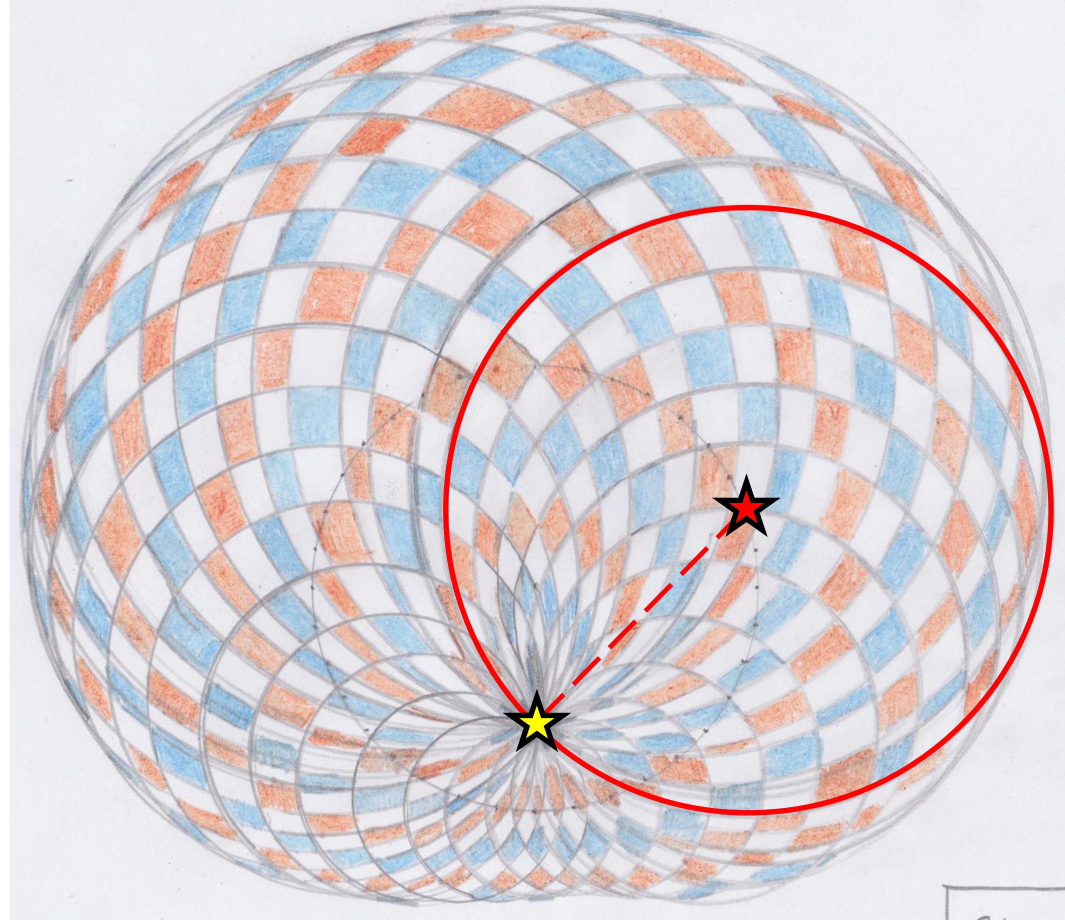
Coding challenge #2:
Cardioid from circles

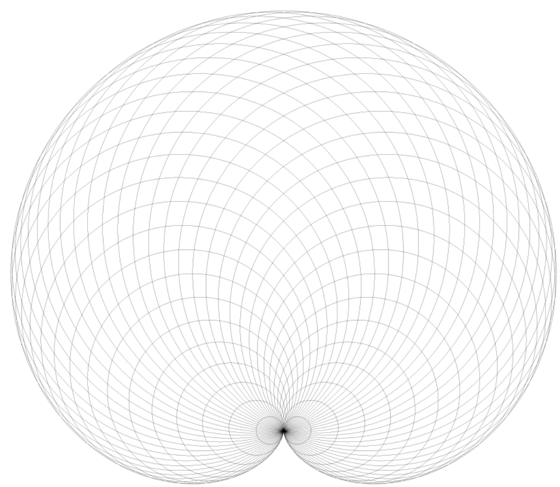
Plot a circle with N points equally spaced along the circumference.

Choose a point inside the circle.

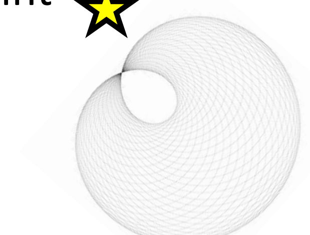
Draw circles with origin at the points along the circle circumference, *that also pass through the chosen point*. This means the circle radius is the distance between the circle circumference point and the chosen point.

Output a PNG file which can be coloured in (see next slide)





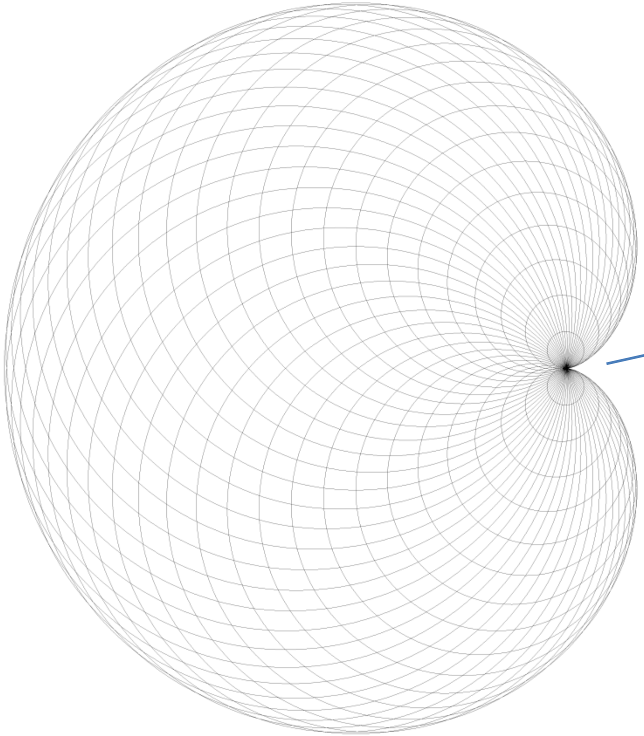
Experiment with moving the 'chosen point'

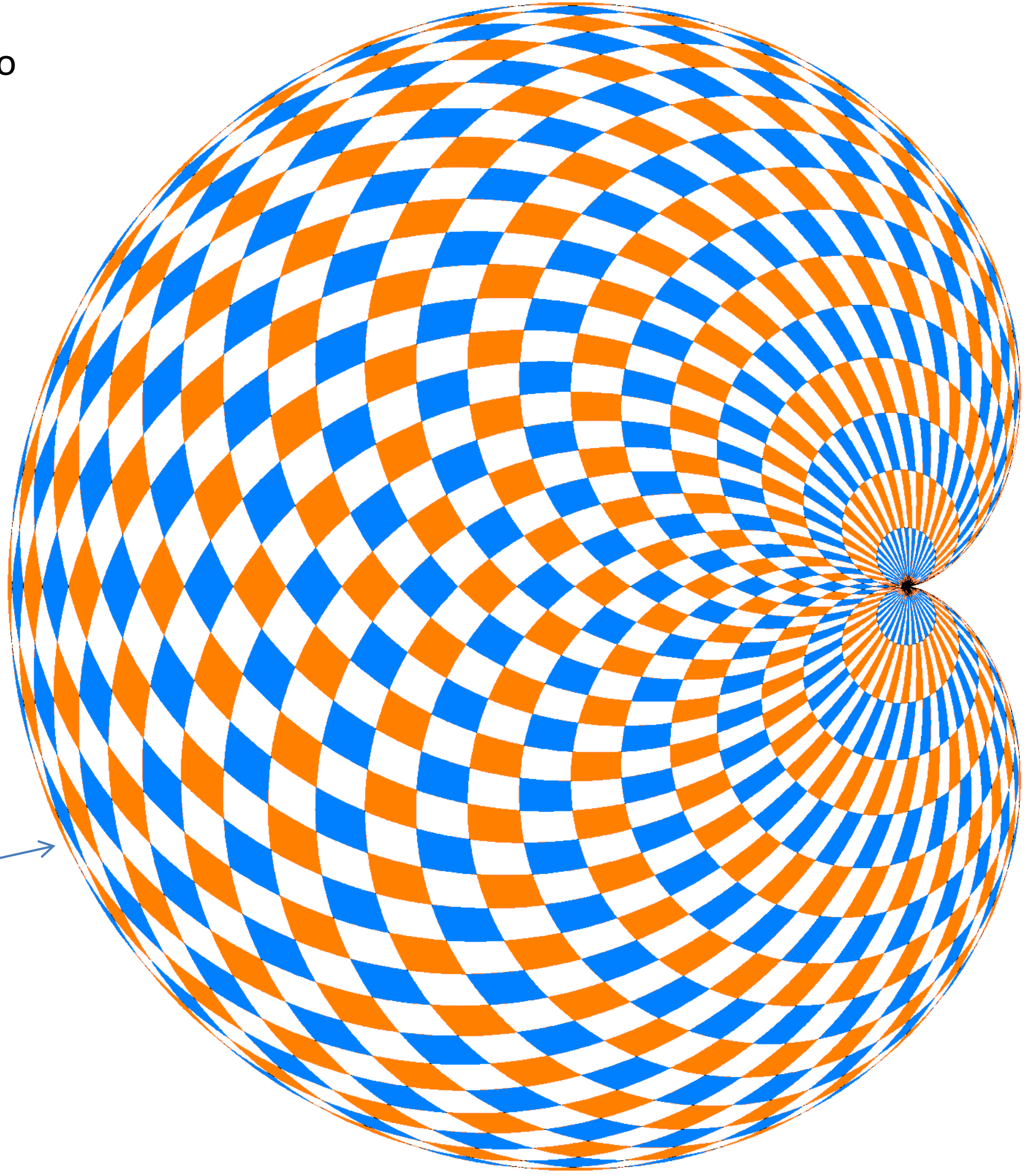


Load your circle construction into a bitmap editor like IrfanView (press F12 to get the editor)

and use the **fill tool** to colour code your image.

It is surprisingly satisfying!





Coding challenge #3: Cardioid curve stitch

Define a ‘clock’ of 100 ‘times’ around a unit circle. The angle between each ‘time’ is 3.6 degrees. Draw lines from each ‘time’ location n round the clock to a time m given by the equation:

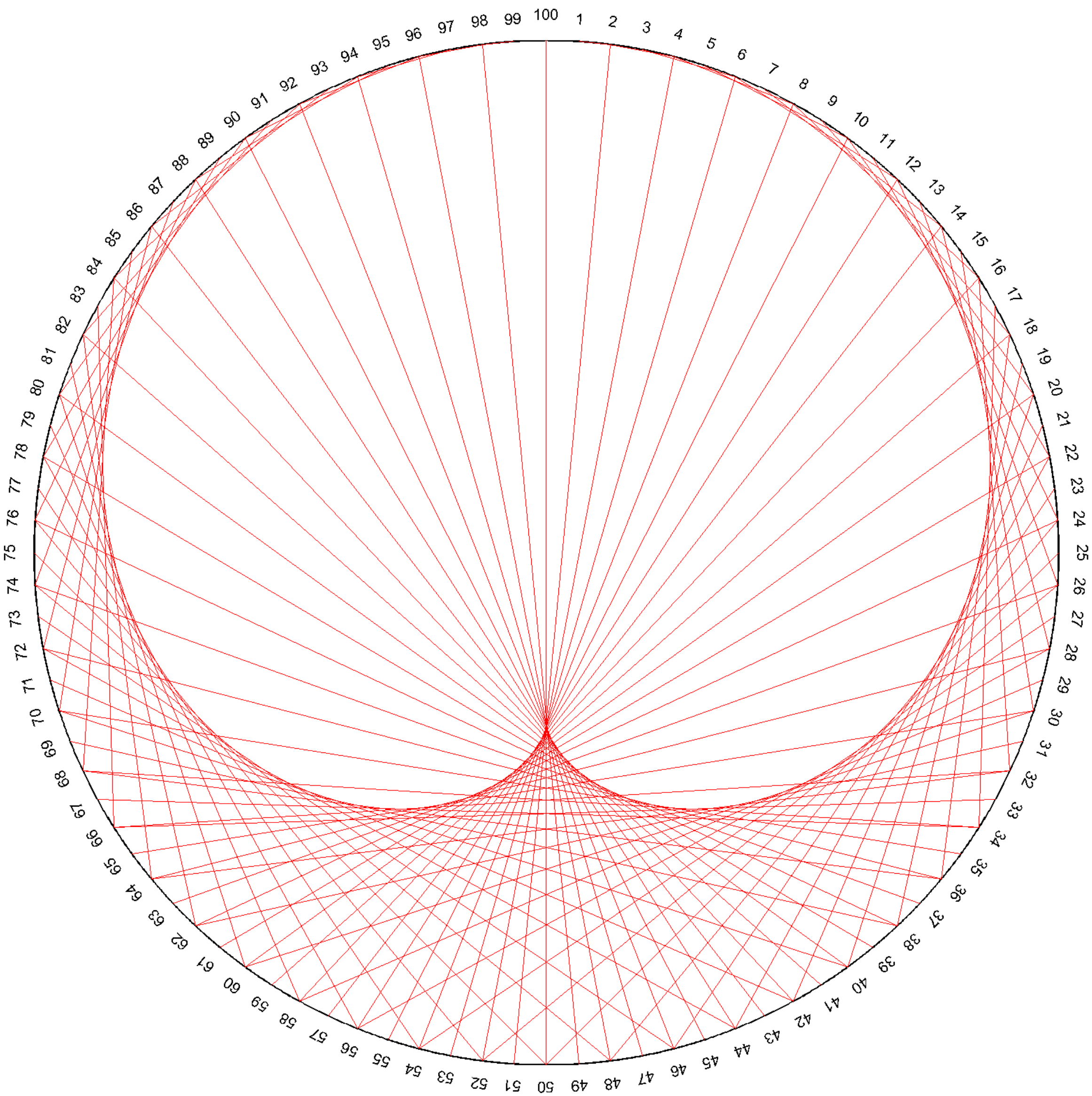
$$m = 2n \mod 100$$

mod means *modulo*, which means ‘subtract whole multiples of 100, and give me the remainder.’

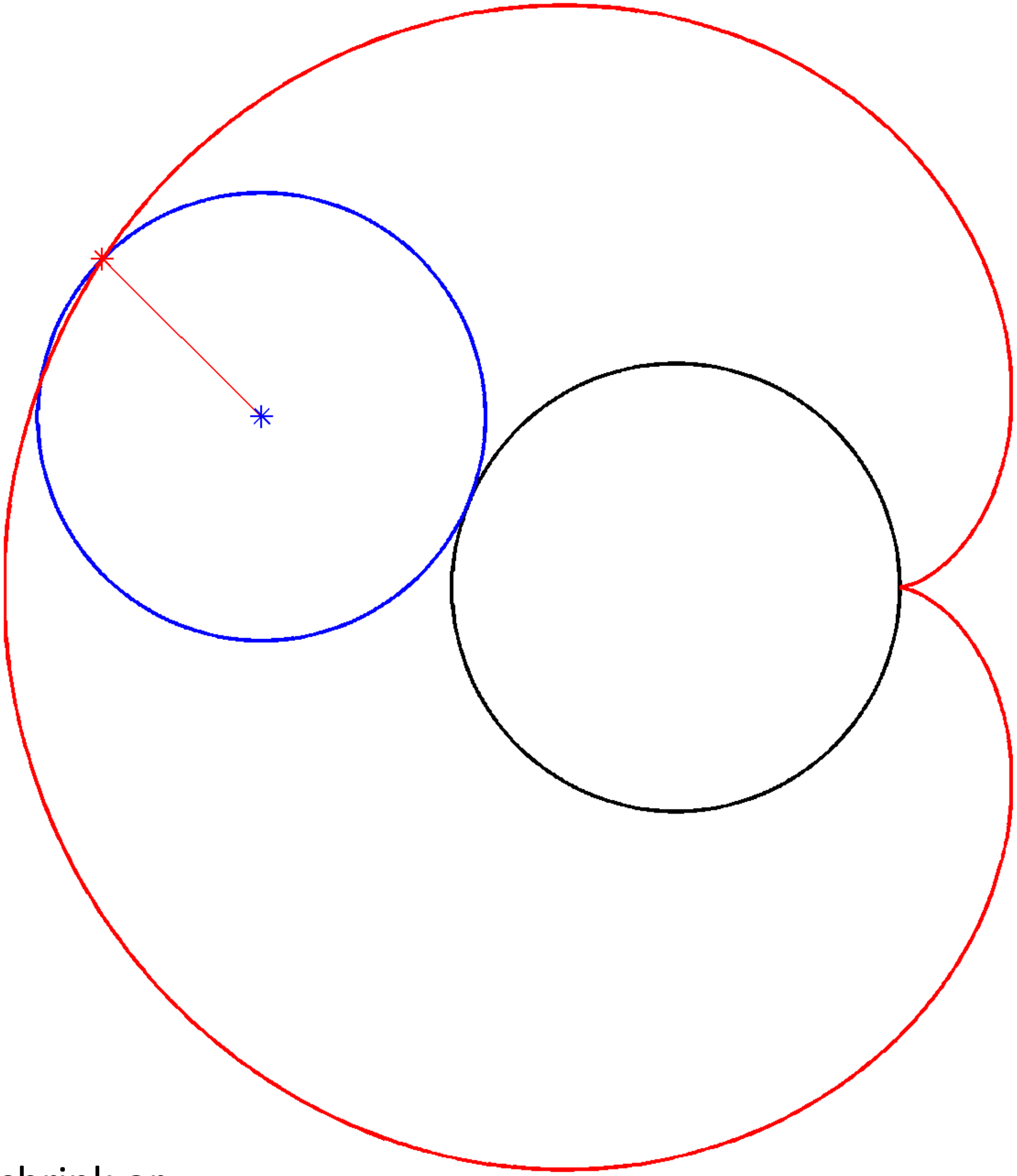
e.g.   105 mod 100 is 5
        317 mod 100 is 17

Eventually your line intersections should form a **cardioid**.

Extension: Use N rather than 100 and make N = 200,500, 1000....



A **cardioid** is the path of a point on a cylinder, that rolls around another identical cylinder, without slipping.



Coding challenge #4:
Rolling cylinders

Construct an *animation* of one circle rolling around another, with a point on the circumference of the outer circle following a cardioid path.

Output is an AVI file or MP4. Upload it to YouTube.

Extension: what happens when you shrink or expand the radius of the outer cylinder? Can you do this *dynamically* when running a program? (e.g. use arrow keys to achieve the change).

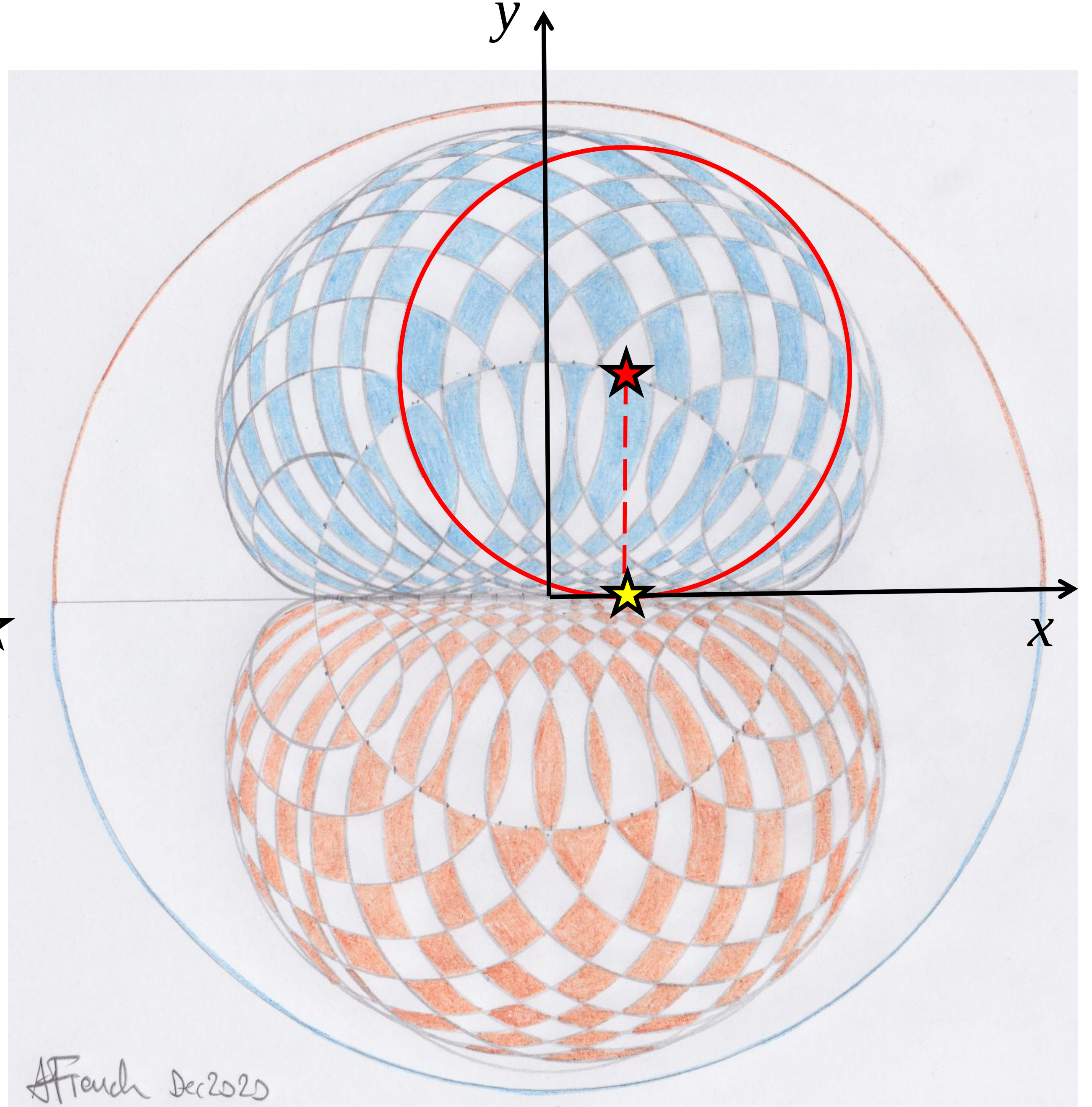
Coding challenge #5:
Nephroid by drawing circles

https://en.wikipedia.org/wiki/Nephroid

Draw a circle and divide the (horizontal) diameter into N equally spaced points.

Work out the vertical coordinates on the circle that correspond to the x coordinate of the diameter points.

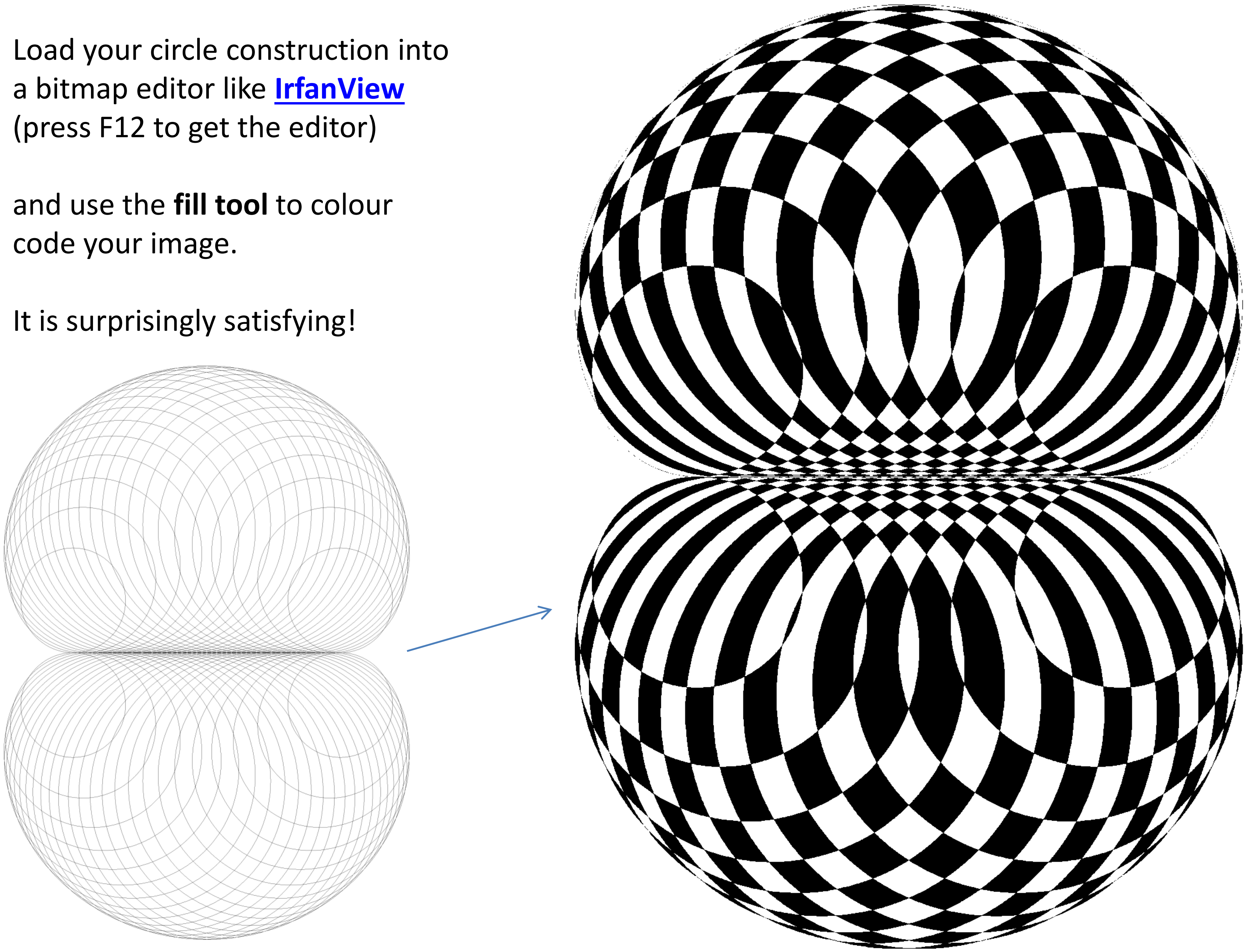
Draw circles, **centred on the circle circumference points**, that all have a radius equal to the *vertical distance* from the circle centre to the horizontal diameter line.



Load your circle construction into a bitmap editor like **IrfanView** (press F12 to get the editor)

and use the **fill tool** to colour code your image.

It is surprisingly satisfying!



Coding challenge #6: Nephroid curve stitch

Define a 'clock' of 100 'times' around a unit circle. The angle between each 'time' is 3.6 degrees. Draw lines from each 'time' location n round the clock to a time m given by the equation:

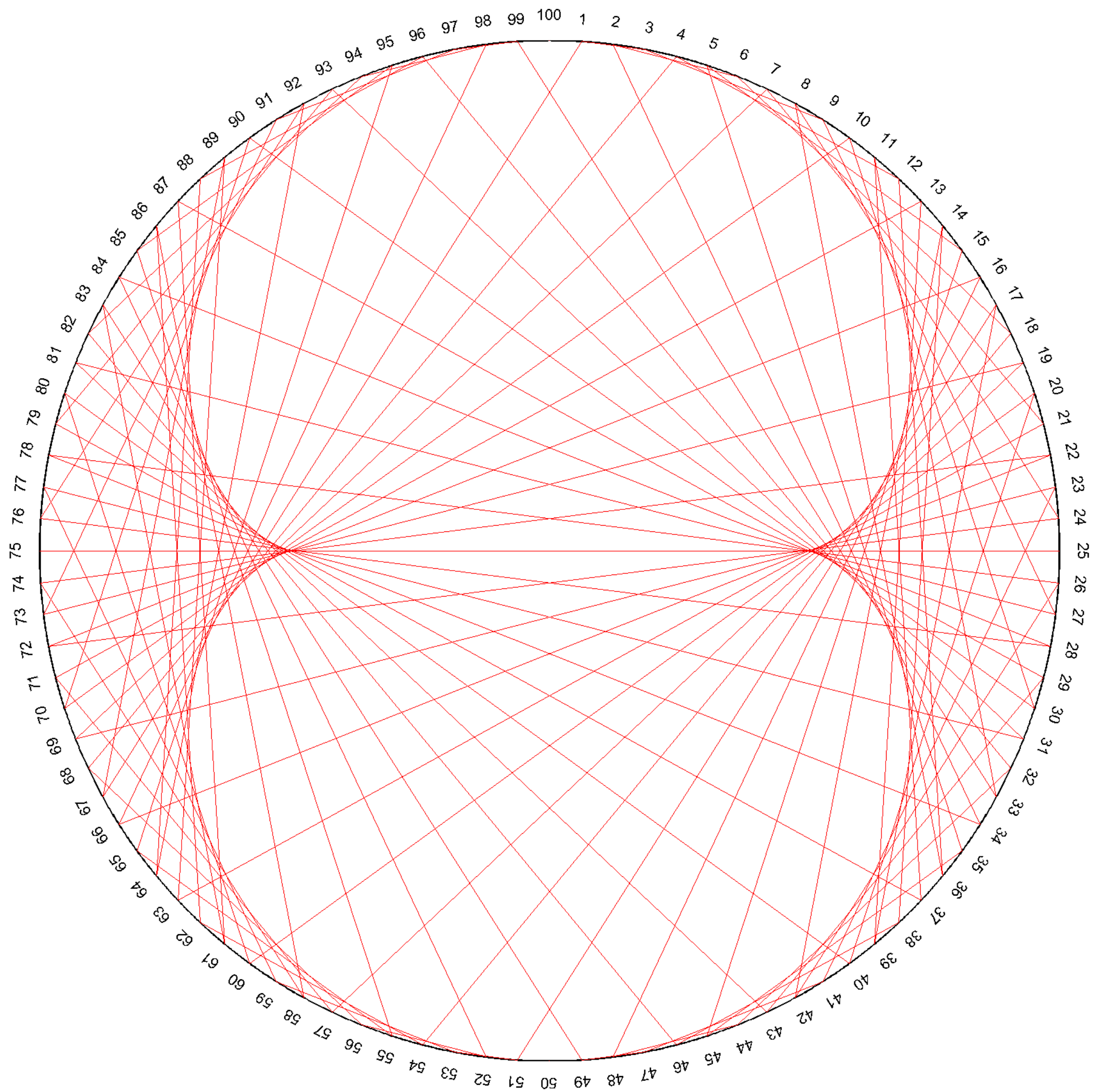
$$m = 3n \mod 100$$

mod means *modulo*, which means 'subtract whole multiples of 100, and give me the remainder.'

e.g. 105 mod 100 is 5
317 mod 100 is 17

Eventually your line intersections should form a **nephroid**.

Extension: Use N rather than 100 and make N = 200,500, 1000....



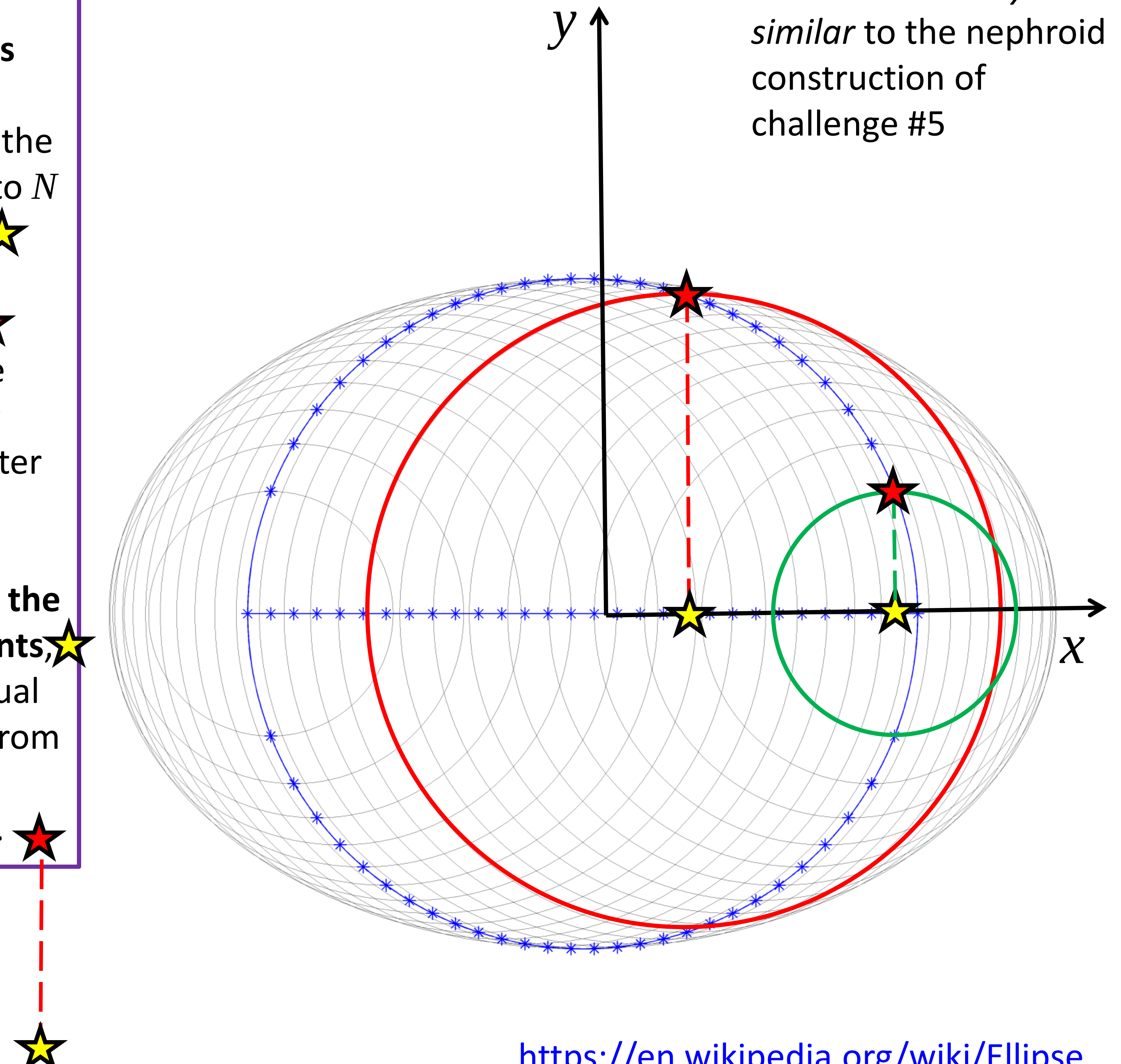
Coding challenge #7:
Ellipse by drawing circles

Draw a circle and divide the (horizontal) diameter into N equally spaced points.

Work out the vertical coordinates on the circle that correspond to the x coordinate of the diameter points.

Draw circles, **centred on the horizontal diameter points**, that all have a radius equal to the *vertical distance* from the circle centre to the horizontal diameter line.

Note – this is *very similar* to the nephroid construction of challenge #5

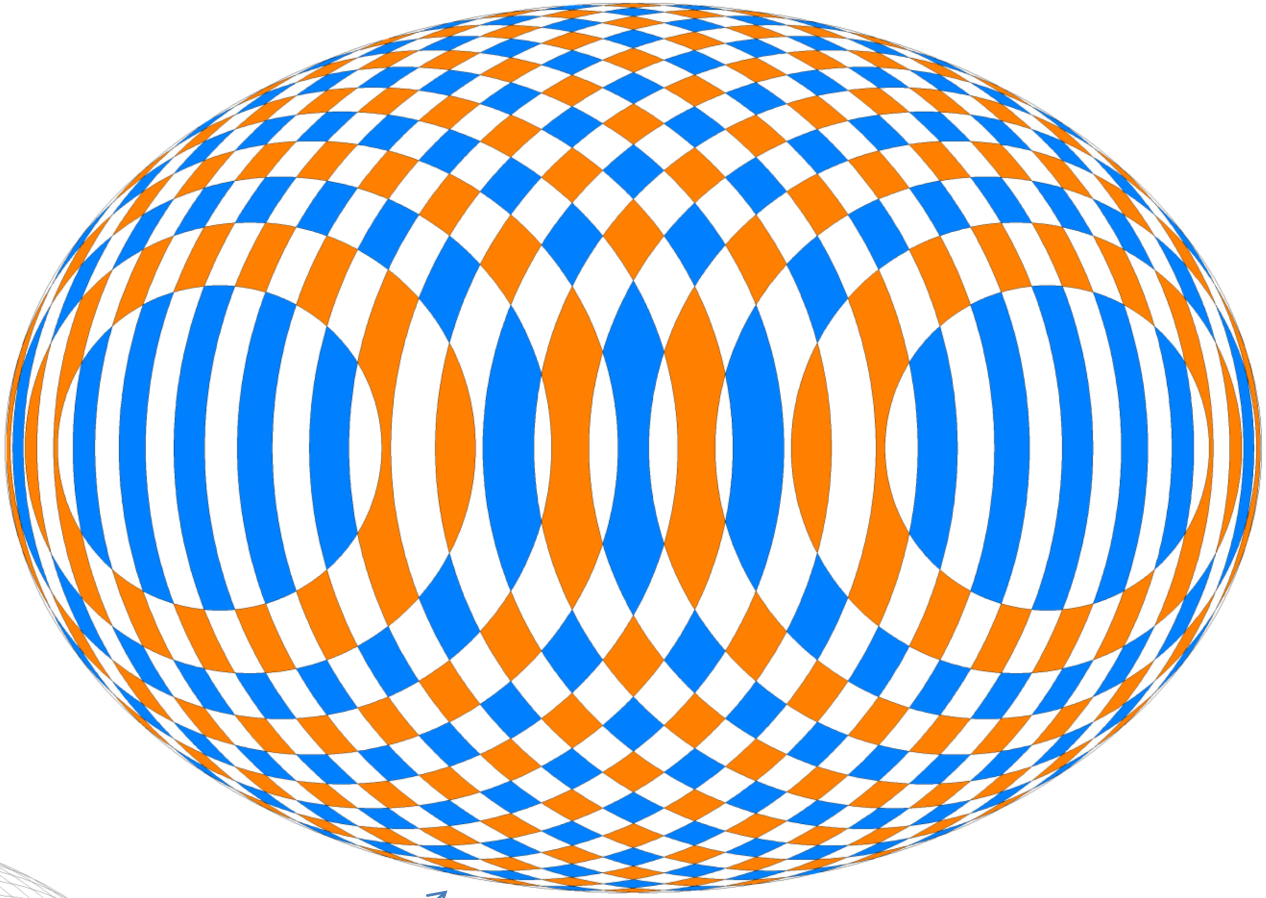


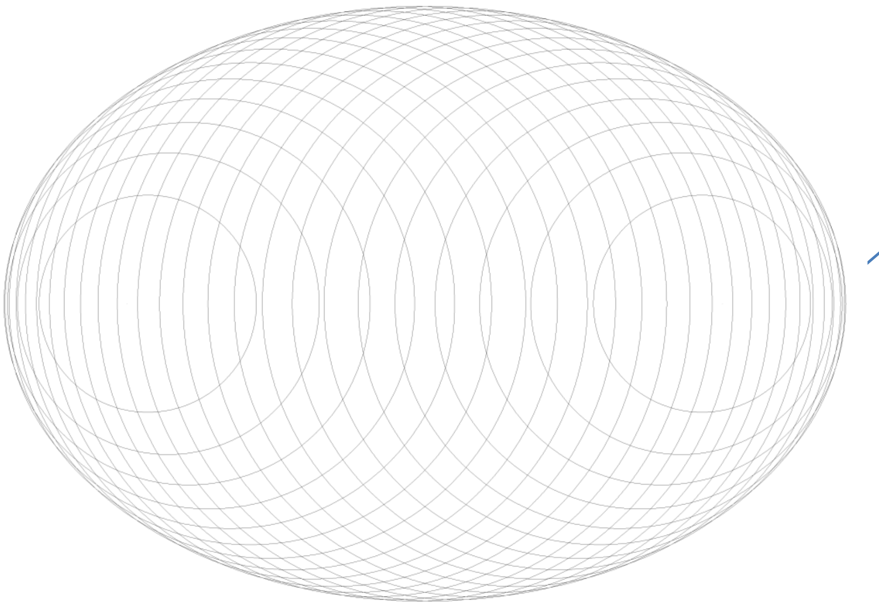
https://en.wikipedia.org/wiki/Ellipse

Load your circle construction i
a bitmap editor like **[IrfanView]**
(press F12 to get the editor)

and use the **fill tool** to colour
code your image.

It is surprisingly satisfying!

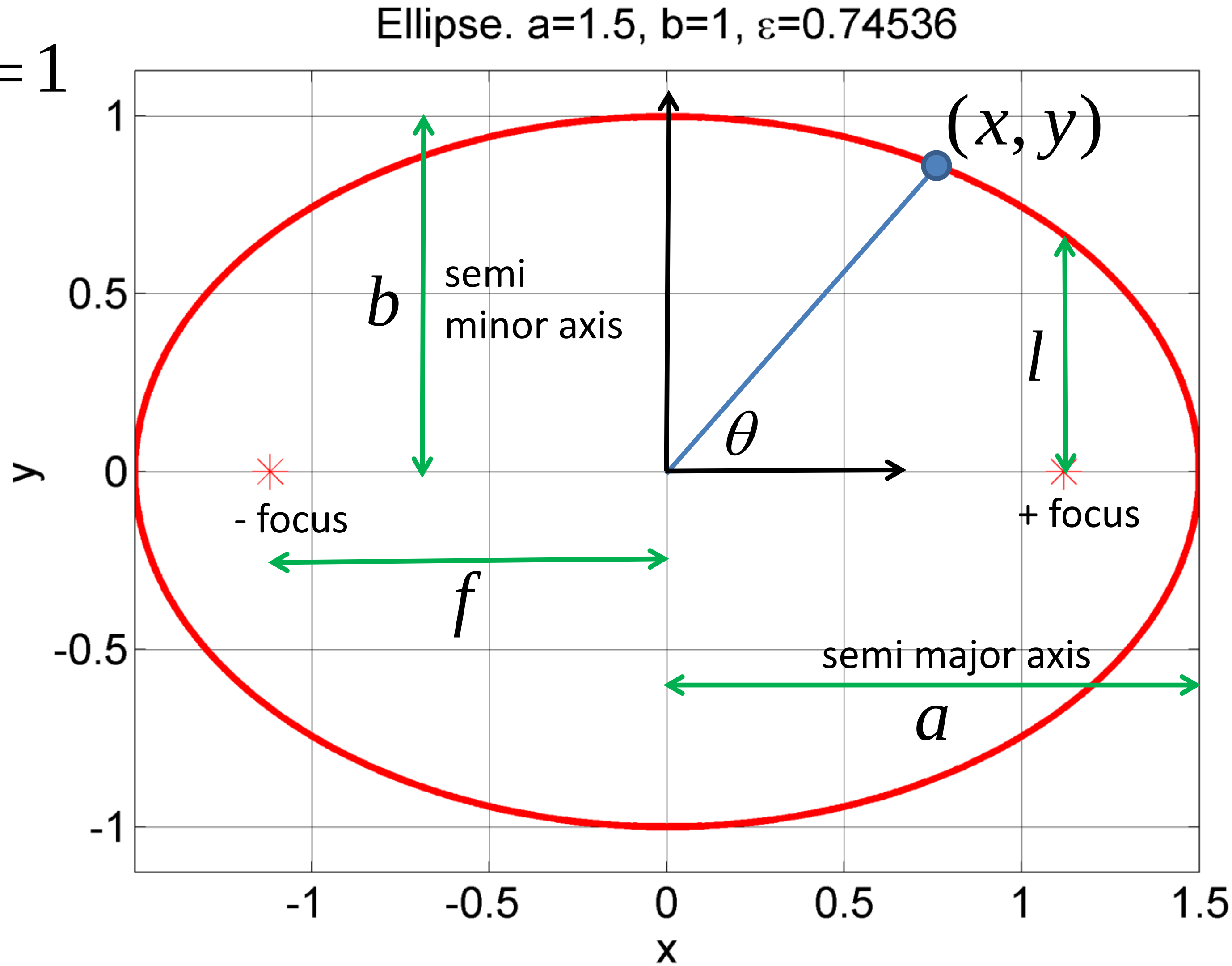




$$\left(\frac{x}{a}\right)^2 + \left(\frac{x}{b}\right)^2 = 1$$

$$x = a\cos\theta$$

$$y = b\sin\theta$$



$$\varepsilon = \sqrt{1 - \frac{b^2}{a^2}}$$

Ellipse **eccentricity**

$$f = \sqrt{a^2 - b^2}$$

Centre to **focus** distance

$$l = b^2 / a$$

Semi latus rectum

https://en.wikipedia.org/wiki/Ellipse

$$r = \frac{a(1-\varepsilon^2)}{1-\varepsilon\cos\theta}$$

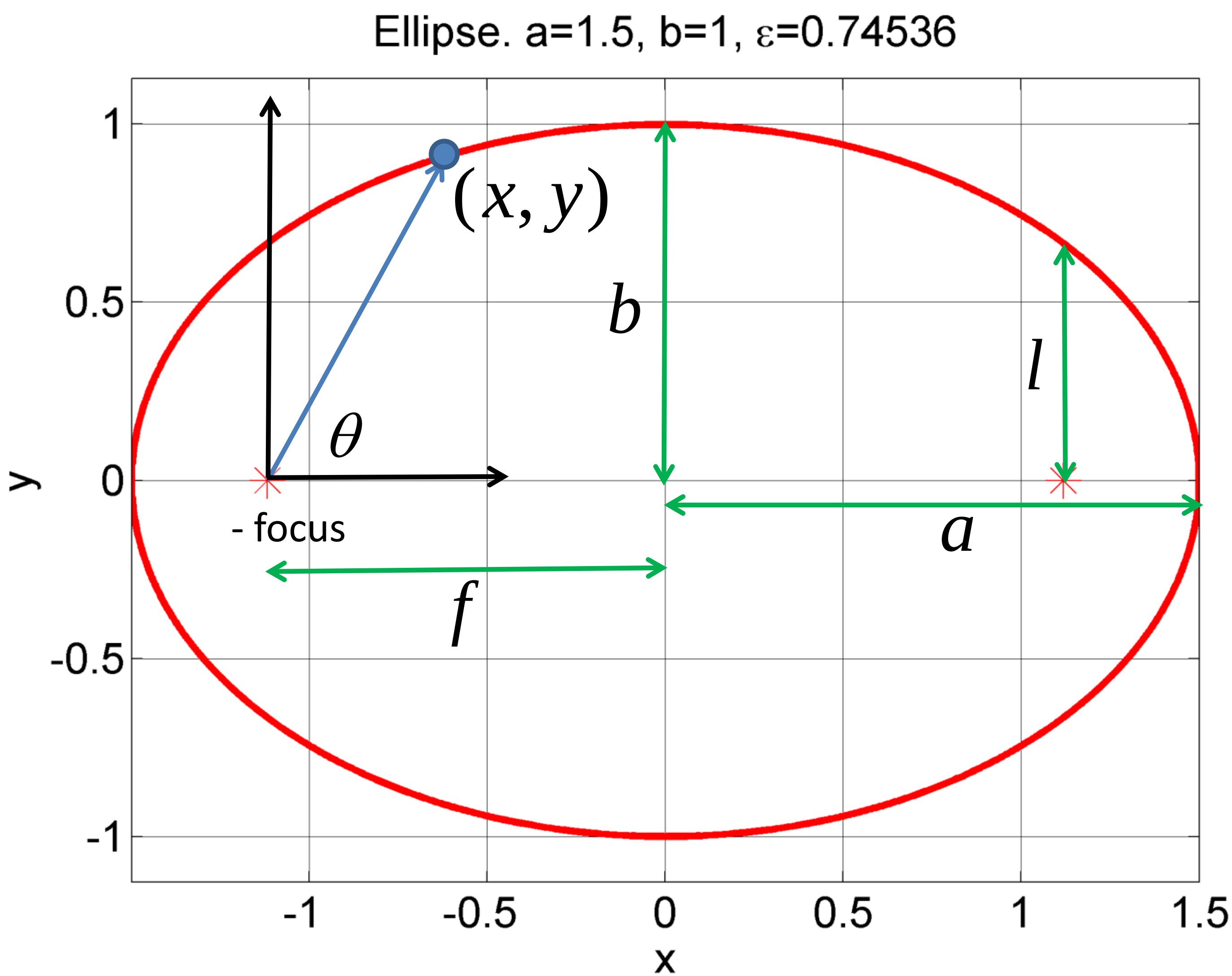
$$r = \frac{l}{1-\varepsilon\cos\theta}$$

$$x = r\cos\theta$$

$$y = r\sin\theta$$

$$\varepsilon = \sqrt{1-\frac{b^2}{a^2}}$$

Ellipse **eccentricity**



$$f = \sqrt{a^2 - b^2}$$

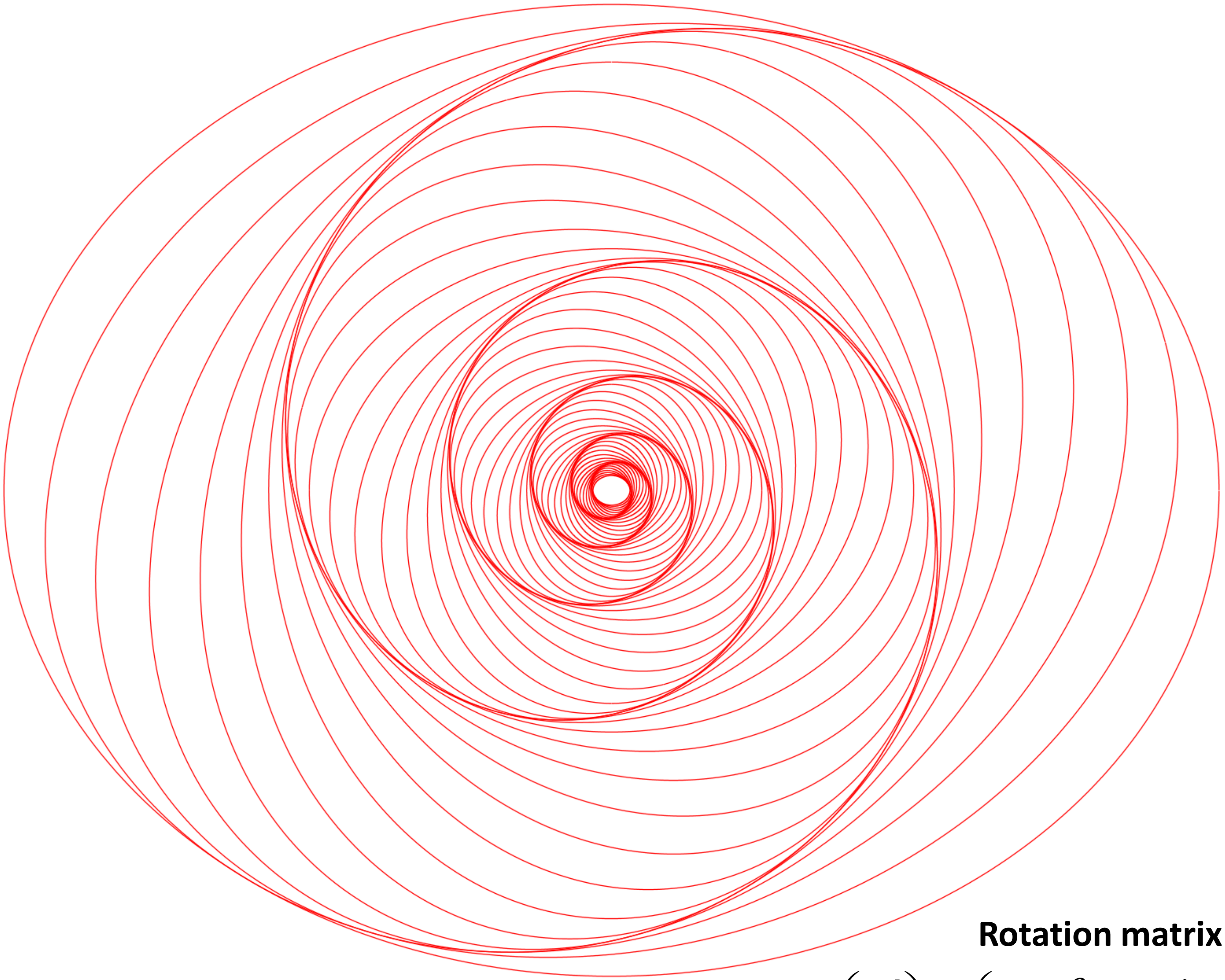
Centre to **focus** distance

$$l = b^2/a$$

Semi latus rectum

Alternative definition using - focus as coordinate centre

https://en.wikipedia.org/wiki/Ellipse



Rotation matrix:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Challenge! Write a program to make a **whorl** based upon ellipses that are *scaled* and then *rotated*.