**Exercise:**

In this question, you are required to draw circles (using structures) according to an algorithm given below.

- Define first four circles with the following features.

| X coordinate | Y coordinate | Color |
|---|---|---|
| 1 | 0 | [1 0 0] |
| 65 | 0 | [1 1 0] |
| 165 | 0 | [0 1 0] |
| 216 | 0 | [0 0 1] |

- Draw first four circles with the given (x, y) coordinates. However, you need to generate integer random numbers (between 5 and 30) for radius of each circle. Print the properties of these circles as the following:

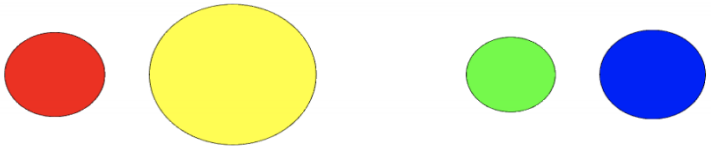
Circle 1: x coordinate:1 y coordinate:0 radius:18

Circle 2: x coordinate:65 y coordinate:0 radius:30

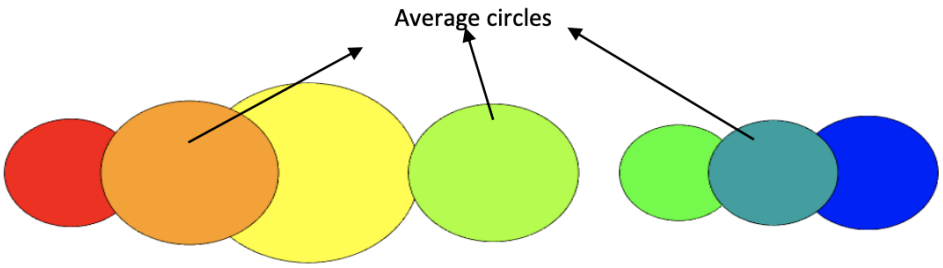
Circle 3: x coordinate:165 y coordinate:0 radius:16

Circle 4: x coordinate:216 y coordinate:0 radius:19

Then, you need to draw these circles. When drawing circles, use function **drawCircle (drawCircle.m** is provided in the submission folder.) An example figure is provided below. You are required to take averages of the coordinates and radii of each consecutive circle pair.



After generating average circles, you can draw these as the following.



- You are required to perform this operation iteratively by choosing two consecutive circles and taking average of them. (*Hint: After each iteration, in order to take averages of two consecutive circles (including the average circles drawn in previous iterations), you need to perform some operations (based on x coordinates of each circle) in each iteration.*).
  You will need to store all of the generated circles in a struct array.
  The loop will stop when difference between centers (x coordinates) of the consecutive circles are less than 1.
- You should title the figure as "Number of generated circles: X"(X will be calculated in the code).
- Note that usage of **structures/struct** is important in this question. Please keep features of your circles in structures. Your code will be graded according to usage of structs, printed output and figure.
- Some useful commands are provided below.

```
close all
figure
axis equal off
hold on
...

hold off
```