1 Explain what, if anything, is wrong with each of the following code fragments. You may assume all variables are initialized appropriately. (3)

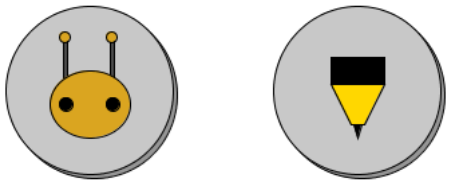
(a)
```
if (a = 4) {
    ellipse(50, 50, 100, 100);
}
```

(b)
```
if (a == 10); {
    ellipse(50, 50, 100, 100);
}
```

(c)
```
if (a == 20)
    ellipse(50, 50, 100, 100);
```

2 Processing, and many other programming languages, have separate operators for comparing the equality of two values (==) and for assigning a value to a variable (=). Briefly discuss the pros and cons of having different operators for these operations. (3)

3 Write a program that will simulate a random coin toss using Processing's `random()` method and display a coin representing either "heads" or "tails". Run your program multiple times to ensure it is working as intended. (5)



4 Write a program that will generate 4 circles of random diameters ranging from 0 to 150 with the following colour requirements: (5)

(a) if the diameter of the circle is less than 50, the circle should be coloured *blue*,

(b) if the diameter of the circle is between 50 and 100 (inclusive), the circle should be coloured *green*, and

(c) if the diameter of the circle is greater than 100, the circle should be coloured *red*.

Run your program multiple times to ensure it is working as intended.

5 The RGB colour model is not the only model used to express different colours in a computer system. Processing also allows us to express colours in a model known as **H**ue, **S**aturation, and **B**rightness (HSB), also sometimes referred to as Hue/Saturation/Value (HSV). Because HSB can be used to express the exact same set of colours as RGB, there is a method for converting between the two. That is, given values for Red, Green, and Blue, you can derive Hue, Saturation, and Brightness using the following algorithm†. (20)

To Calculate Hue:

1. First, calculate Rp, Gp, and Bp such that:

$$Rp = R \div 255$$
$$Gp = G \div 255$$
$$Bp = B \div 255$$

2. Next, calculate max, min, and d as:

$$max = max(Rp, Gp, Bp)$$
$$min = min(Rp, Gp, Bp)$$
$$d = max - min$$

3. Finally, calculate H using the following piecewise function:

$$H = \begin{cases} 0, & d = 0 \\ 60 \times \left(\frac{Gp - Bp}{d} + 6\right), & max = Rp \\ 60 \times \left(\frac{Bp - Rp}{d} + 2\right), & max = Gp \\ 60 \times \left(\frac{Rp - Gp}{d} + 4\right), & max = Bp \end{cases}$$

†This algorithm is adapted from the RapidTables RGB to HSV conversion algorithm found at: http://www.rapidtables.com/convert/color/rgb-to-hsv.htm.

To Calculate Saturation:
Calculate S using the following piecewise function:

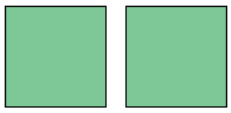
$$S = \begin{cases} 0, & max = 0 \\ \frac{d}{max}, & max \neq 0 \end{cases}$$

To Calculate Brightness:
The brightness is always equal to the highest of Rp, Gp, and Bp. That is,

$$Br = max$$

Note: This will result in an H value between 0 and 360 and S and Br values between 0 and 1. Processing allows us to switch to this colour model using its `colorMode()` method. In particular, to switch to the values generated using the above algorithm, we can call: `colorMode(HSB, 360, 1, 1)`.

Write a program that initializes values for `R`, `G`, and `B` and uses the above algorithm to calculate `H`, `S`, and `Br`. Test your program by displaying two squares, side-by-side, the first using a `fill()` in RGB and the second using a `fill()` in HSB (after having set the `colorMode()`).



Note: Vary R, G, and B between runnings of your program in order to ensure H, S, and Br are being calculated correctly.

6 *Complementary colours* are colours that appear on opposite sides of the colour wheel. Although the colour wheel was first developed by artists using primary pigment colours (red, blue, and yellow), we can still find a colour complementary to one using the RGB model. In order to do so, it is important to note that the range of *hue* values in the HSB model being $0–360$ is no coincidence: hue actually refers to the angle around the colour wheel at which the particular colour appears. The algorithm for finding an RGB modelled colour's complement, therefore, is: (10)

To Find an RGB Complement:

1. Convert from RGB to HSB.
2. Calculate $H = (H + 180) \bmod 360$.
3. Convert from HSB to RGB.

Note: Because Processing allows us to draw in HSB colour mode, the last step of the algorithm is not strictly required and is not part of this assigment.

(a) Write a program that initializes values for `R`, `G`, and `B` and uses the above algorithm (omitting Step #3) to find the colour's complement. Test your program by displaying two squares, side-by-side, the first using a `fill()` of the original colour and the second using a `fill()` of its complement. Make sure you switch to the HSB `colorMode()` before drawing the second square.
Hint: Use your RGB to HSB program from Question #5.

(b) Explain why adding 180 to H in the HSB value finds a colour on the opposite side of the colour wheel.

(c) Explain why the new H value needs to be taken $\bmod 360$.