1. String concatenation: Complete the `println` instruction so that the output of

   ```java
   for (int i = 0; i < 10; i++) {
     int num = (int) random(100);
     println("i = " + i + ": The random number is " + num); // finish
   }
   ```

2. For the function below:
   a. What is the return type?  float
   b. What is the name of the parameter?  red
   c. What is the type of the parameter?  int
   d. Which of the following are legal ways (or reasonable) to call the function?
      i. `float r = convertRed();`  // bad – no parameter
      ii. `int r = convertRed(2.5);` // bad – wrong type for return and parameter
      iii. `float r = convertRed(155);` // ok
      iv. `convertRed(100);` // syntax ok but nothing done with return value
      v. `stroke(convertRed(100), 1.0, 1.0));` // ok

3. To set a color in Processing, you use the command `stroke(r,g,b)`. To set a grayscale value, you just use a single number `stroke(g)` where `g` can be computed from the RGB value by adding together 30% of the red value, 59% of the green value, and 11% of the blue value. For example, if `r=10` (out of 255), `g =100`, and `b = 255`, then the grayscale value will be `g = (0.3*10)+(0.59*100)+((0.11*255) = 90.2`. Write a function that takes the three rgb integer values as parameters, and returns the grayscale value as a float.

   ```java
   float calcGray(int r, int g, int b) {
     float gray = 0.3*r + 0.59*g + 0.11*b;
     return gray;
   }
   ```
4. Complex numbers: Place the following in standard form \(a + bi\).
   a. \(i^3\) ______ -i _____________
   b. \(\sqrt{-36} + 3i^2\) ______ -3 + 6 i______________

5. Complex numbers: Given \(z_1 = -1 + 7i\) and \(z_2 = (2 + i)\). Calculate the following, placing the result in standard form
   a. \(z_1 + z_2 = \) ___1 + 8 i__
   b. \(z_1 - z_2 = \) ___-3 + 6 i______
   c. \(2z_1\) ___-2 + 14 i_______
   d. \(z_1z_1 = z_1^2 = \) ___-48 – 14 i_______
   e. \(z_1z_2 = \) ___-9 + 13 i___
   f. \(\overline{z}_1 + z_1 = \) ___-2 _________
   g. \(\overline{z}_1 z_1 = \) ___50_________
   h. Length of \(z_1 = |z_1| = \) __\(\sqrt{50}\) ____________

6. Class syntax: In class, we made use of a Complex class in Processing to compute the Mandelbrot set.
   a. How would you create a new Complex object with real component equal to 1.5 and imaginary component equal to -6?
      \(\text{Complex} \ c = \text{new Complex}(1.5, \ -6);\)

   b. In Processing, suppose you have created complex numbers \(c_1, c_2, \text{ and } c_3\). How do you multiply \(c_1\) and \(c_2\) together, placing the result in \(c_3\)?
      \(c_3 = \text{Complex.cMult}(c_1, c_2);\)

   c. In Processing, suppose you have created complex numbers \(c_1, c_2, \text{ and } c_3\). How do you compute (i.e. what is the syntax of) for computing
      \(c_3 = c_1*c_2 + c_1\)
      \(c_3 = \text{Complex.cAdd( Complex.cMult(c1, c2) , c1 );}\)
What is the standard form for the complex numbers whose values in polar coordinates are

d. \( r = 2, \theta = 90^\circ \) \( 2i \)
e. \( r = 1, \theta = 180^\circ \) \(-1\)

7. What is the polar coordinate representation \((r, \theta)\) for the following complex numbers

a. \(-3i\) \((r, \theta) = (3, 270)\)
b. \(1+i\) \((r, \theta) = (\sqrt{2}, 45)\)

8. Rescaling: Given a region of the complex plane where the real component ranges between -10 and 20, and the imaginary part ranges between 15 and 35 as shown in the figure:

```
int pixeli = map(5, -10, 20, 0, width);
int pixelj = map(21, 15, 35, 0, height);
```

b. How does one use the map function to determine the complex number corresponding to the pixel \((i,j)\)
```
int real = map(i, 0, width, -10, 20);
int imag = map(j, 0, height, 15, 35);
```

9. Convert the following for-loop to a while loop:
```
for (int i = 0; i < 100; i++) {
    println(i);
}
```
```
int i = 0;
while (i < 100) {
    println(i);
    i++;
}
10. Mandelbrot Set: Write a do-while loop that iterates on the complex function \( z = z^2 + c \). It stops when either the loop has iterated 100 times or the length of \( z \) exceeds 2. Initialize \( z \) and \( c \) to be \( z = 0 \) and \( c = 0.5 + i \).

```java
int k = 0;
do {
    z = Complex.cAdd( Complex.cMult(z, z) , c );
k++;
} while ( k < 100 && Complex.len(z) < 2);
```

11. Recursion: Write a recursive function to output the numbers from 0 to 100 in reverse order.

```java
void setup() {
    printNums(100);
}

void printNums(int n) {
    if (n < 0) {
        return;
    }
    else {
        println(n);
        printNums(n-1);
    }
}
```

12. Recursion: Write a recursive function to add the numbers from 1 to \( n \), for some value of \( n \).

```java
void setup() {
    int n = 100;
    println ("The sum from 1 to "+ n + " is "+ addNums(n));
}

int addNums(int n) {
    if (n <= 0) {
        return 0;
    }
    else {
        return n + addNums(n-1);
    }
}
```