Worksheet: Final Review 2

1. Consider the following method. There are a number of syntax errors in the code.

```
1 public static circumference(double radius) {
2    return 2 x math.pi x radius
3 }
```

a) Rewrite the method, correcting all the syntax errors.

b) Write a Java statement that calls the method `circumference` with the parameter `radius` set to `2.0`, and prints the result that is returned by the method.

2. Consider the following method. Refer to the line numbers to make your answers more clear.

```
 1 public static boolean inArray(int[][] a, int value) {
 2    for(int row = a.length-1; row > 0; row--) {
 3       for(int col = 0; col <= a[row].length; col++) {
 4          if( a[row][col] == value ) {
 5             return true;
 6          }
 7       }
 8    }
 9    return false;
10 }
```

a) Explain why the code will sometimes throw an `ArrayIndexOutOfBoundsException`.

b) Explain why the code will sometimes return `false`, even when `value` is present in array `a`.

Worksheet: Final Review 2

3. A method that takes a single parameter of type String, is intended to return the same String with all characters removed that are at even indices. The partial method is given below. Complete the method.

```
public static String getOddLetters(String input) {
   String output = "";
   for(                                                    )
   {
      output += input.substring(                );
   }
   return output;
}
```

4. Consider the following code segment, and the fact that $\cos\left(\frac{\pi}{3}\right) = \sin\left(\frac{\pi}{6}\right) = 0.5$.

```
double eq1 = Math.cos(Math.PI / 3);
double eq2 = Math.sin(Math.PI / 6);
System.out.println(eq1 + ", " + eq2);
System.out.println(eq1 == eq2);
```

The output of the above code is found to be as follows.

```
0.5000000000000001, 0.49999999999999994
false
```

This is due to rounding error during computation. The solution to this is to avoid directly comparing floating point values that may be subject to rounding errors using the double equals operator (`==`). We instead check if the difference between the two values is below a threshold. Given a `double` value named `threshold`, write an expression that will check if the difference between `eq1` and `eq2` is less than `threshold`.

5. The method below requires three parameters, `calc`, `base`, and `pow`.

```
public static double myMethod( /* missing parameter list */ ) {
   if(calc == true) {
      for(int i = pow; i > 1; i--) {
         base *= base;
      }
   }
   return base;
}
```

Write the code to replace ***/* missing parameter list */*** such that the method call `myMethod(true, 0.5, 2)` will run without error and return the value `0.25`.