Chapter 2 Exercises

2.1.1. Write a static method max3() that takes three int arguments and returns the value of the largest one. Add an overloaded function that does the same thing with three double values.

```
public class Max3Calculator {
   public static int max3(int a, int b, int c){
      int max = a;
      if (b > max) max = b;
      if (c > max) max = c;
      return max;
   }
   public static double max3(double a, double b, double c){
      double max = a;
      if (b > max) max = b;
      if (c > max) max = c;
      return max;
   }

   public static void main(String[] args) {
      System.out.println("Max of (3, 7, 5):" + max3(3, 5, 7));
      System.out.println("Max of (2.5, 3.1, 1.8,): " + max3(2.5, 3.1, 1.8));
   }
}
```

2.1.2 Write a static method odd() that takes three Boolean arguments and returns true if an odd number of the argument values are true, and false otherwise.

```
public class OddChecker {
   public static boolean odd(boolean a, boolean b, boolean c){
      int count = ( a ? 1: 0) + (b ? 1: 0) + ( c ? 1 : 0);
      return count %2 == 1;
   }

   public static void main(String[] args) {
      System.out.println("odd(true, true, false): " + odd(true, true, false));
      System.out.println("odd(true, false, true): " + odd(true, false, true));
      System.out.println("odd(false, false, false): " + odd(false, false, false));
      System.out.println("odd(true, true, true): " + odd(true, true, true));
   }
}
```

2.1.3 Write a static method majority() that takes three boolean arguments and returns true if at least two of the argument values are true, and false otherwise. Do not use an if statement.

```
public class MajorityChecker {

   public static boolean majority(boolean a, boolean b, boolean c) {

      return ((a ? 1 : 0) + (b ? 1 : 0) + (c ? 1 : 0)) >= 2;
   }


   public static void main(String[] args) {

      System.out.println("majority(true, true, false): " + majority(true, true, false));
      System.out.println("majority(true, false, false): " + majority(true, false, false));
      System.out.println("majority(false, false, false): " + majority(false, false, false));
      System.out.println("majority(true, true, true): " + majority(true, true, true));
   }
}
```

2.1.4 Write a static method eq() that takes two int arrays as arguments and returns true if the arrays have the same length and all corresponding pairs of of elements are equal, and false otherwise.

```
public class ArrayEqualityChecker {
   public static boolean eq(int[] array1, int[] array2) {

      if (array1.length != array2.length) {
         return false;
      }


      for (int i = 0; i < array1.length; i++) {
         if (array1[i] != array2[i]) {
            return false;
         }
      }


      return true;
   }


   public static void main(String[] args) {

      int[] arr1 = {1, 2, 3};
      int[] arr2 = {1, 2, 3};
      int[] arr3 = {1, 2, 4};
      int[] arr4 = {1, 2};

      System.out.println("eq(arr1, arr2): " + eq(arr1, arr2));
      System.out.println("eq(arr1, arr3): " + eq(arr1, arr3));
      System.out.println("eq(arr1, arr4): " + eq(arr1, arr4));
   }
}
```

2.1.5 Write a static method areTriangular() that takes three double arguments and returns true if they could be the sides of a triangle (none of them is greater than or equal to the sum of the other two).

```
public class TriangleChecker {
  public static boolean areTriangular(double a, double b, double c) {

    return (a + b > c) && (a + c > b) && (b + c > a);
  }


  public static void main(String[] args) {

    System.out.println("Are 3.0, 4.0, 5.0 triangular? " + areTriangular(3.0, 4.0, 5.0));
    System.out.println("Are 1.0, 2.0, 3.0 triangular? " + areTriangular(1.0, 2.0, 3.0));
    System.out.println("Are 6.0, 10.0, 15.0 triangular? " + areTriangular(6.0, 10.0, 15.0));
    System.out.println("Are 7.0, 24.0, 25.0 triangular? " + areTriangular(7.0, 24.0, 25.0));
  }
```