1. What is the output from the following code fragment:
   ```java
   for (int i = 0; i < 100; i++) {
       if (i < 10 || i % 2 == 0) {
           continue;
       }
       if (i == 17) {
           break;
       }
   System.out.println(i);
   }
   ```

2. Implement a class named Dog. A dog should have all of the following private data members: name (String) and a license number (int) and an age (int). The Dog should have all of the following methods - also write junit tests to make sure all of the methods work as described.

   //a constructor with no parameters that creates a dog named "Snoopy" with license 1 and age 0
   //a constructor with one String parameter that creates a dog with the passed named with license 1 & age 0.
   //a constructor with 3 parameters that creates a dog with all of the values passed
   // implement 3 getters and 3 setters for all of the individual data members of the Dog
   //also add a toString method that returns a string of all of the values (e.g. Snoopy(1): age 0)
   //also add an equals method - two dogs are equal if their names and license numbers match
   // then add the following:
   public void birthday();//increments the age of the dog
   public void addPet();//Modifies the name of the Dog to have "-pet" after it
   //extra challenge: add "-pet" only if the "-pet" is not already there as the end of the name

3. Consider the following code fragment:
   ```java
   int[] a = {7, 9, 26};
   f(a[1]);
   g(a);
   ```
   a. Is it possible that the call to f will modify an element in the array a? **YES / NO**
   b. Is it possible that the call to g will modify an element in the array, a? **YES / NO**
   c. Consider also how this relates to the more general case of primitives and objects that are passed as parameters.

4. Implement a static method called revEq that takes in two Strings as its two parameters. It returns true if the first string equals the second reversed. In other words "Jan" and "nai" would return true,

5. Write a method that will implement "subset". A subset requires that every element in the first array must also be in the second array. You may assume neither array contains any duplicate entries. You may write a private helper method if you would find it useful (but you must also implement that method if you want it).
   ```java
   public static boolean subset(int[] set1, int[] set2);
   /* returns true if and only if every element of set1 also appears in set2*/
   ```

6. Write a method that will implement "proper subset". A proper subset requires that every element in the first array must also be in the second array but there is at least one element of the second array which is not in the first array. You may assume neither array contains any duplicate entries. You may write a private helper method or use the previous method if you would find it useful .
   ```java
   public static boolean properSubset(int[] set1, int[] set2);
   /* returns true if and only if every element of set1 also appears in set2 but there is at least one member of set2 which is not present in set1 */
   ```

7. Redo each of the previous two questions this time passing sets of Stirngs rather than sets of ints.
8. Explain Determine the output for the following program:
   public class ExceptionQuestion {
      private static void f(int val) {
         System.out.println("In F");
         int x = 5/val;  // Division by zero could happen
         System.out.println("Done F");
      }
      public static void main(String[] args) {
         try {
            f(1);
            System.out.println("After 1");
            f(0);
            System.out.println("After 0");
         } catch (ArithmeticException e) {
            System.out.println("AE caught");
         } catch (NullPointerException e) {
            System.out.println("NP caught");
         } finally {
            System.out.println("Finally Done");
         }
      }
   }

9. Explain the difference between the String and the StringBuffer and how that difference can be noticed when passing an object of that type as a parameter to a method. (Be sure to include an explanation of when there wouldn't be any difference.)

10. Implement a static boolean method that takes in a string and returns true if the string contains the same character more than once and false if all characters in the string are unique. Write a junit test to verify that this method is correct (you may use the String library methods - you do not have to memorize these but it is good to be comfortable using them).

11. Implement a method that takes in an array of integers and counts the number of odd values in the array. It returns that count as the return value of the method. (Note: no other parameters are passed.)

12. Implement a method that takes in an array of integers and a single integer as the two parameters. It counts the number of values in the array that are larger than the single integer passed. It returns that count as the return value of the method.

13. Implement a main method that uses an initializer to put values into an array of integers, prints the number of odd values (using the method written for the question above) and then prints the number of even values (also using that method). It then creates two arrays one that will hold only the even values and the other that will hold only the odd values. It then fills those two arrays by copying the correct types of values into each of the two you created. It then prints all odds on one line and all evens on the next line.

14. Modify the main method above so that instead of creating two new arrays, the original array is modified to have all of its even values removed. (You may assume there is at least one odd and one even in the original, but you have to make sure that you aren't wasting any space in that array.)