3. Write code that creates an array of integers named `data` of size 5 with the following contents:

```
public class makeAnArray {
    public static void main(String[] args) {
        int[] intArray = new int[]{ 27,51,33,-1,101 };
        System.out.println(intArray);
    }
}
```

4. Write code that stores all odd numbers between -6 and 38 into an array using a loop. Make the array's size exactly large enough to store the numbers.

```
public static void main(String[] args) {
          Scanner in = new Scanner(System.in);
          int a = in.nextInt();
          int b = in.nextInt();
          int[] intArray = new int[(a+b)/2];
          int index = 0;
          for (int i = -6; i <= 38; i++) {
               if (i % 2 == 1 || -i % 2 == 1) {
                    intArray[index] = i;
                    index++;
               }
               else {
                    ;
               }
          }
     }
```

5. What elements does the array numbers contain after the following code is executed?

```
[0,4,11,0,44,0,0,2,0]
```

6. What elements does the array `data` contain after the following code is executed?

```
[3,3,0,0,6,0,9,-18,0]
```

7. What is wrong with the following code?

```
int[] first = new int[2];
first[0] = 3;
first[1] = 7;
int[] second = new int[2];
second[0] = 3;
second[1] = 7;

// print the array element
System.out.println(first);
System.out.println(second);

// see if the elements are the same
if (first == second) {
    System.out.println("They contain the same elements.");
} else {
    System.out.println("The elements are different.");
}
```

The System.out.println command prints each array's hashcode which won't work in this instance. You should use

```
System.out.println(Arrays.toString(first)) // same thing for second
```

10. Write a piece of code that examines an array of integers and reports the maximum value in the array. Consider putting your code into a method called `max` that accepts the array as a parameters and returns the maximum value. Assume that the array contains at least one element.

```
import java.util.Scanner;
public class test {
     public static void main(String[] args) {
          Scanner sc = new Scanner(System.in);
          System.out.println("Size?");
          int size = sc.nextInt();
          int[] myArray = new int[size];
```

```
          System.out.println("Enter an element, and then press enter. Repeat. ");
          for(int i=0; i<size; i++) {
               myArray[i] = sc.nextInt();
            }
            test m = new test();
            System.out.println(m.max(myArray));

     }
     public int max(int [] array) {
          int max = 0;
          for(int i=0; i<array.length; i++ ) {
             if(array[i]>max) {
                max = array[i];
             }
          }
          System.out.println(max);
          return max;

       }
}
```

11. Write a method called `average` that computes the average (arithmetic mean) of all elements in an array of integers and returns the answer as a `double`. For example, if the array passed contains the value `[1, -2, 4, -4 9, -6, 16, -8, 25, -10]`, the calculated average should be `2.5`. Your method accepts an array of integers as its parameter and returns the average.

```
import java.util.Scanner;
public class test {

     public static double average(int[] array){
          double num = 0;
          for(int index = 0;index<=array.length-1;index++) {
               num = num + array[index];
          }

          double solution = num/(array.length);
          double actual = solution;
          return actual;
     }
     public static void main(String[] args) {
          int[] intArray = new int[]{1, -2, 4, -4, 9, -6, 16, -8, 25, -10};
          System.out.println(average(intArray));
     }
}
```