Directions: This is a closed book, closed notes midterm. Place your answers in the space provided. The point value for each question is indicated. There are a total of 60 points that will be weighted to 100 points. You have 55 minutes for this midterm.

1. (8 pts) Consider the following method.

```java
public static void ifElseMystery(int x, int y) {
    int z = 4;
    if (z <= x) {
        z = x + 1;
    } else {
        z = z + 9;
    }
    if (z <= y) {
        y++;
    }
    System.out.println(z + " " + y);
}
```

What is the output of the following four calls?

a. `ifElseMystery(3, 20);`

b. `ifElseMystery(4, 5);`

c. `ifElseMystery(5, 5);`
2. (9 pts) The following method is supposed to return the number of factors of a given integer \( n \). For example, `countFactors(12)` should return 6 because there are 6 values that divide 12 evenly: 1, 2, 3, 4, 6, and 12. Is this method written correctly? If so, answer yes. If not rewrite the code so that it works as intended.

```java
public static int countFactors (int n) {
    for (int i = 1; i <= n; i++) {
        if (n % i == 0) {
            return i;
        }
    }
    return 0;
}
```
3. (10 pts) Consider the following code. What range of values can each variable \((a, b, c, d,\) and \(e)\) have? Specify ranges with a dash i.e. "1 - 10" and separate multiple numbers with commas i.e. "1,2,3,4,5,etc..."

```java
Random rand = new Random();
int a = rand.nextInt(100);
int b = rand.nextInt(20) + 50;
int c = rand.nextInt(20 + 50);
int d = rand.nextInt(100) - 20;
int e = rand.nextInt(10) * 4;
```

4. (9 pts) Write a method called `printLetters()` that accepts a `String` as its parameter and uses a fencepost loop to print the letters of the string, separated by commas. For example, the call `printLetters("Rabbit");` should print the following output:

```
R, a, b, b, i, t
```
5. (15 pts) For each of the five points labeled by comments, identify each of the assertions in the table below as either being always true, never true, or sometimes true / sometimes false.

```java
public static int mystery(Scanner console, int x) {
    int y = console.nextInt();
    int count = 0;

    // Insert assertions here.
}
```
// Point A
while (y < x) {
    // Point B
    if (y == 0) {
        count++;
        // Point C
    }
    // Point D
    y = console.nextInt();
}

// Point E
return count;

<table>
<thead>
<tr>
<th>Point</th>
<th>y &lt; x</th>
<th>y == 0</th>
<th>count &gt; 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point E</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. (9 pts) Consider a file called *readme.txt* that has the following contents:

What would be the output produced by the following code segment when it is run on the *readme.txt* file?

```java
Scanner input = new Scanner(new File("readme.txt"));
int count = 0;
while (input.hasNextLine()) {
    System.out.println("input: " + input.nextLine());
    count++;
}
System.out.println(count + " total");
```