Problem #1:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 * (2 + 4) - 3 * 5</td>
<td>9</td>
</tr>
<tr>
<td>54 % 10 + 8 * 3 % 9</td>
<td>10</td>
</tr>
<tr>
<td>3 * 2 + 4 + &quot;+&quot; + 2 + 3 * 4</td>
<td>&quot;10+212&quot;</td>
</tr>
<tr>
<td>2.3 * 3 + 19 / 5 / 2 + 6.0 / 5</td>
<td>9.1</td>
</tr>
<tr>
<td>108 / 20 * 3 / 4 / 2.0 + 1.0 / 2</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Problem #2:

to walk the walk is good

to hear the good is bad

to feel the walk is song

to feel the talk is bad

Problem #3:

<table>
<thead>
<tr>
<th>Method Call</th>
<th>Output Produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifElseMystery(14, 14);</td>
<td>14 13</td>
</tr>
<tr>
<td>ifElseMystery(4, 5);</td>
<td>7 5</td>
</tr>
<tr>
<td>ifElseMystery(10, 5);</td>
<td>12 10</td>
</tr>
<tr>
<td>ifElseMystery(2, 8);</td>
<td>3 8</td>
</tr>
<tr>
<td>ifElseMystery(5, 4);</td>
<td>5 9</td>
</tr>
<tr>
<td>ifElseMystery(6, 8);</td>
<td>7 8</td>
</tr>
</tbody>
</table>

Problem #4:

<table>
<thead>
<tr>
<th>Method Call</th>
<th>Output Produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>mystery(8);</td>
<td>1 8</td>
</tr>
<tr>
<td>mystery(32);</td>
<td>2 5</td>
</tr>
<tr>
<td>mystery(184);</td>
<td>3 13</td>
</tr>
<tr>
<td>mystery(8239);</td>
<td>4 22</td>
</tr>
</tbody>
</table>

Problem #5:

<table>
<thead>
<tr>
<th></th>
<th>n == 0</th>
<th>n % 2 == 1</th>
<th>x == 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point A</td>
<td>sometimes</td>
<td>sometimes</td>
<td>always</td>
</tr>
<tr>
<td>Point B</td>
<td>never</td>
<td>sometimes</td>
<td>sometimes</td>
</tr>
<tr>
<td>Point C</td>
<td>never</td>
<td>always</td>
<td>never</td>
</tr>
<tr>
<td>Point D</td>
<td>sometimes</td>
<td>sometimes</td>
<td>sometimes</td>
</tr>
<tr>
<td>Point E</td>
<td>always</td>
<td>never</td>
<td>sometimes</td>
</tr>
</tbody>
</table>
Problem #6

Solution:

```java
public static boolean printThreeDigit(Random r, int n) {
    boolean seen = false;
    System.out.print("numbers: ");
    if (n > 0) {
        int number = r.nextInt(900) + 100;
        System.out.print(number);
        if (number == 777) {
            seen = true;
        }
        for (int i = 2; i <= n; i++) {
            number = r.nextInt(900) + 100;
            System.out.print(",", number);
            if (number == 777) {
                seen = true;
            }
        }
    }
    System.out.println();
    if (seen) {
        System.out.println("We saw a 777");
    } else {
        System.out.println("no 777");
    }
    return seen;
}
```

Problem #7

Solution:

```java
public static void findRange(Scanner console) {
    System.out.print("number (0 to quit)? ");
    int number = console.nextInt();
    int max = number;
    int min = number;
    while (number != 0) {
        if (number > max) {
            max = number;
        } else if (number < min) {
            min = number;
        }
        System.out.print("number (0 to quit)? ");
        number = console.nextInt();
    }
    int range = max - min;
    System.out.println("range = " + range);
}
```
Problem #8

Two Possible Solutions:

```java
public static void printStripped(String s) {
    boolean inComment = false;
    for (int i = 0; i < s.length(); i++) {
        char next = s.charAt(i);
        if (next == '<') {
            inComment = true;
        } else if (inComment && next == '>') {
            inComment = false;
        } else if (!inComment) {
            System.out.print(next);
        }
    }
    System.out.println();
}

public static void printStripped(String s) {
    int start = s.indexOf('<');
    while (start != -1) {
        int stop = s.indexOf('>', start + 1);
        s = s.substring(0, start) + s.substring(stop + 1);
        start = s.indexOf('<');
    }
    System.out.println(s);
}
```