Q1: Give the value and type of each of the following Java expressions. If an expression will not compile or will cause an exception at runtime, put an X under value. If the value is a string, enclose it in double quotes.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / 0</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>&quot;800&quot; * 1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>&quot;1&quot; + &quot; - &quot; + &quot;1&quot;</td>
<td>&quot;1-1&quot;</td>
<td>String</td>
</tr>
<tr>
<td>3.14159 + (int) Math.PI</td>
<td>6.14159</td>
<td>double</td>
</tr>
<tr>
<td>1-1-1-1</td>
<td>-2</td>
<td>int</td>
</tr>
<tr>
<td>3 / 2.0 + 2 * 5</td>
<td>11.5</td>
<td>double</td>
</tr>
<tr>
<td>(8 &lt;= 2)</td>
<td></td>
<td>(2e8 &lt;= 8e2)</td>
</tr>
<tr>
<td>Double.parseDouble(&quot;8.5*2&quot;)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>&quot;1&quot; + 1 + 1 + &quot;1&quot;</td>
<td>&quot;1111&quot;</td>
<td>String</td>
</tr>
</tbody>
</table>
Q2: Consider the following code:

```java
public class MethodTester {
    private static void methodB(int[] c, int d) {
        c[0]++;
        d += 42;
    }
    private static int methodA(int[] a, int b) {
        methodB(a, b);
        a[0]++;
        return b/2;
    }
    public static void main(String[] args) {
        int[] arr = {8, 9, 10};
        int x = 1;
        x = methodA(arr, x);
        System.out.println(arr[0] + " " + x);
    }
}
```

Which one of the following is the output of this program? Circle your answer.

- 8 3
- 8 10
- 8 21
- 9 1
- 9 3
- 9 21
- 10 0
- 10 1
- 10 21
Q3: Here is a method that draws squares recursively:

```java
public static void draw(int n, double x, double y, double r) {
    if (n==0) return;  // base case
    draw(n-1, x, y, r/4);
    StdDraw.square(x, y, r);  // draw a square
    draw(n-1, x - r/2, y, r/4);
    draw(n-1, x + r/2, y, r/4);
}
```

Below, we plot the picture produced when `draw(3, 0.5, 0.5, 0.5)` is called. It draws thirteen squares, which we have also labelled with dashed circles and arrows. What is the order in which the squares were drawn? Write all of the integers from 1 to 13 in the circles to indicate this order, with 1 labelling the first square drawn and 13 the last.
Q4: Recursion: Consider the following program:

```java
public class Series {
    public static int func(int j) {
        if (j==1) return 1;
        return 2 * func(j - 1) + 5 * func(j - 2);
    }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]); // assume N >= 0
        System.out.println(func(N));
    }
}
```

a. Draw the recursion tree for \(\text{func}(3)\). You only need to draw the tree up to 3 levels, which means the height of the recursion tree should be no greater than 3.

```
  func(3)
     /\  \\
    2 * func(2) + 5 * func(1)
       /\   /
    2 * func(1) + 5 * func(0)
       /\   /
  2 * func(-1) + 5 * func(-2)
```

b. From the recursion tree in (a), do you see a problem with the program? Explain what is the problem.

The problem is the function reductive step skips over the base case which will result in a stack overflow error.

Change if (j==1) return 1; to if (j <= 1) return 1;