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 and type. 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>X</td>
</tr>
<tr>
<td>&quot;800&quot; * 1</td>
<td>X</td>
<td>X</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>X</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?

"8 3"
"8 10"
"8 21"
"9 1"
"9 3"
"9 21"
"10 0"
"10 1"
"10 21"
Q3: Consider the following code:

```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 `func(3)`. You only need to draw the tree up to 3 levels.

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

b. What is the problem with this recursive program, based on the tree you drew above?

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;
Q4: Fill in the blanks in the following table.

<table>
<thead>
<tr>
<th>hex</th>
<th>decimal</th>
<th>16-bit two's complement</th>
<th>TOY instruction pseudo-code</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFE</td>
<td>-2</td>
<td>1111111111111110</td>
<td>R[F] = PC; PC = FE</td>
</tr>
<tr>
<td>77FF</td>
<td>30,719</td>
<td>0111011111111111</td>
<td>R[7] = 00FF</td>
</tr>
<tr>
<td>FF01</td>
<td>-255</td>
<td>1111111110000001</td>
<td>R[F] = PC; PC = 01</td>
</tr>
<tr>
<td>7A00</td>
<td>31232</td>
<td>0111101000000000</td>
<td>R[A] = 0000</td>
</tr>
</tbody>
</table>

Hint: \(7 \cdot 16^3 = 28,672\).
**Q5:** Consider the following TOY program:

\[
\begin{array}{l}
20: \; 81FF \; R[1] = \text{stdin} \\
21: \; \text{SEE BELOW} \\
23: \; 0000 \quad \text{halt}
\end{array}
\]

What is the value of \(R[1]\) after executing the code above, where \(M[21]\) is replaced by one of the following instructions. Note, the PC starts on line 20 and 1111 is on standard input.

Your answers must be four hex digits.

<table>
<thead>
<tr>
<th>(M[21] \text{ set to})</th>
<th>(\text{Value of } R[1] \text{ after halt})</th>
</tr>
</thead>
<tbody>
<tr>
<td>21: 0000 (\text{halt})</td>
<td>1111</td>
</tr>
<tr>
<td>21: 7100 (R[1] = 0000)</td>
<td>0000</td>
</tr>
<tr>
<td>21: C023 (\text{PC = 23})</td>
<td>1111</td>
</tr>
<tr>
<td>21: 9122 (M[22] = R[1])</td>
<td>2222</td>
</tr>
</tbody>
</table>