COS 126 **Mini Practice Exam Solutions** **30** min

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.

Expression	Value	Type
`1 / 0`	**X**	**X**
`"800" * 1`	**X**	**X**
`"1" + " - " + "1"`	**"1 - 1"**	**String**
`3.14159 + (int) Math.PI`	**6.14159**	**double**
`1-1-1-1`	**-2**	**int**
`3 / 2.0 + 2 * 5`	**11.5**	**double**
`(8 <= 2) || (2e8 <= 8e2)`	**false**	**boolean**
`Double.parseDouble("8.5*2")`	**X**	**X**
`"1" + 1 + 1 + "1"`	**"1111"**	**String**

Q2: Consider the following code:

```
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"

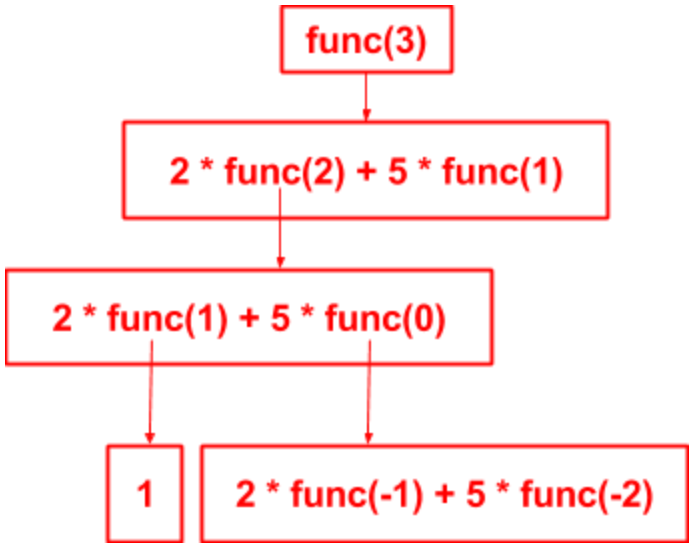
Q3: Consider the following code:

```
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.



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

The problem is the 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.

hex	decimal	16-bit two's complement	TOY instruction pseudo-code
FFFE	-2	1111111111111110	`R[F] = PC; PC = FE`
1234	4,660	**0001001000110100**	`R[2] = R[3] + R[4]`
1101	4,353	0001000100000001	**`R[1] = R[0] + R[1]`**
77FF	30,719	0111011111111111	`R[7] = 00FF`
FF01	**-255**	**1111111100000001**	**`R[F] = PC; PC = 01`**
7A00	**31232**	0111101000000000	**`R[A] = 0000`**

Hint: $7 \cdot 16^3 = 28{,}672$.

Q5: Consider the following TOY program:

```
20: 81FF  R[1] = stdin
21:     SEE BELOW
22: 1211  R[2] = R[1] + R[1]
23: 0000  halt
```

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.

`M[21]` set to	Value of `R[1]` after halt
`21: 1111  R[1] = R[1] + R[1]`	**2222**
`21: 0000  halt`	**1111**
`21: 1211  R[2] = R[1] + R[1]`	**1111**
`21: 7100  R[1] = 0000`	**0000**
`21: C023  PC = 23`	**1111**
`21: 8121  R[1] = M[21]`	**8121**
`21: 9122  M[22] = R[1]`	**2222**