1. **Feedback**. Complete the survey linked from the moodle after completing this assignment. Any non-empty answer will receive full credit.

2. **Scala Basics**: Binding and Scope. For each the following uses of names, give the line where that name is bound. Briefly explain your reasoning (in no more than 1 - 2 sentences).

   (a) Consider the following Scala code.

```
1    val pi = 3.14
2    def circumference(r: Double): Double = {
3      val pi = 3.14159
4      2.0 * pi * r
5    }
6    def area(r: Double): Double =
7      pi * r * r
```

   The use of pi at line 4 is bound at which line? The use of pi at line 7 is bound at which line?

   The use of pi at line 4 is bound at line 3, where it is declared in a local scope, so that will be used rather than the global definition.

   The use of pi at line 7 is bound at line 1, where it is defined globally and there are no definitions within the area function.

(b) Consider the following Scala code.

```
1   val x = 3
2   def f(x: Int): Int =
3     x match {
4       case 0 => 0
5       case x => {
6         val y = x + 1
7         ({
8           val x = y + 1
9           y
10         }*f(x-1))
11      }
12    }
13  val y = x + f(x)
```

The use of x at line 3 is bound at which line? The use of x at line 6 is bound at which line? The use of x at line 10 is bound at which line? The use of x at line 13 is bound at which line?

The use of x at line 3 is bound at line 2
The use of x at line 6 is bound at line 5
The use of x at line 10 is bound at line 5
The use of x at line 13 is bound at line 1

3. **Scala Basics:** Typing. In the following, I have left off the return type of function g. The body of g is well-typed if we can come up with a valid return type. Is the body of g well-typed?

```
1    def g(x: Int) = {
2      val (a, b) = (1, (x, 3))
3      if (x == 0) (b, 1) else (b, a + 2)
4    }
```

The corrected version would be:

```
1    def g(x: Int): ((Int,Int), Int) = {
2      val (a, b) = (1, (x, 3))
3      if (x == 0) (b, 1) else (b, a + 2)
4    }
```

If so, give the return type of g and explain how you determined this type. For this explanation, first, give the types for the names a and b.

Explanation:

$$(b, a) : ((Int, Int), Int)\ because \tag{1}$$
$$a : Int\ because \tag{2}$$
$$x\ =\ int \tag{3}$$
$$b : (Int, Int)\ because \tag{4}$$
$$x = int, 3 = int \tag{5}$$
$$\tag{6}$$

Stop when you reach values (or names). As an example of the suggested format, consider the plus function:

```
1    def plus(x: Int, y: Int) = x + y
```

Yes, the body expression of plus is well-typed with type Int.

$$x + y : Int\ because \tag{7}$$
$$x : Int \tag{8}$$
$$y : Int \tag{9}$$