CSE 3302/5307 Lab Assignment 1

Due October 13, 2015

Goals:

Understanding of Scheme and elementary functional programming concepts, especially the three types of "eaters".

Requirements:

1. Write the following four Scheme functions:

   a. `(bitCount` num`)` - counts the number of one bits in the binary representation of a non-negative integer.

   b. `(numCount` lat num`)` - counts the number of positions in a lat that match num, a non-negative integer.

   c. `(numCountExp` x num`)` - counts the number of positions in an S-expression that match num, a non-negative integer.

   d. `(structurally?` x y`)` - check whether two S-expressions are structurally identical. They are allowed only to differ in the types of atoms that appear.

2. Submit your Racket source file on Blackboard by 3:15 p.m. on Tuesday, October 13.

```
> (bitCount 0)
0
> (bitCount 1)
1
> (bitCount 10)
2
> (bitCount 65)
2
> (bitCount 32)
1
> (bitCount (- 32 1))
5
> (numCount '() 0)
0
> (numCount '(1 2 3 a 5 b 6 c "string" 7 5 9) 5)
2
> (numCount '(5 b 6 c "string" 7 5 1 2 3 a 9) 0)
0
> (numCount '(1  c "string" 7 5 2 3 a 5 b (6 (1 18 4) 3) 9) 1)
1
> (numCountExp '(1 2 (3 a 5) (b 6 c "string" 7 (5)) 9) 5)
2
> (numCountExp '(1 2 (3 a 5) (b 6 c ("string") 7 (5)) 9) 0)
0
> (numCountExp '(1 2 (3 a 5) 1 (b 6 () 1 c ("string") 7 (5)) 9) 1)
3
```

```
> (structurally? '(1 2 (3 a 5) (b 6 c "string" 7 (5)) 9)
                 '(2 1 (3 "string" 5) (b 6 c a 7 (5)) 9))
#t
> (structurally? '(1 2 (3 a b 5) (b 6 c "string" 7 (5)) 9)
                 '(2 1 (3 "string" 5) (b 6 c d a 7 (5)) 9))
#f
> (structurally? '(a b c d) '(b c d e f))
#f
> (structurally? 'a 1234)
#t
> (structurally? '(()) '())
#f
```

Getting Started:

1. Nested helper functions are encouraged.

2. Error checking is not expected.

3. `set!` is not needed.