Lab 0 – Update
(9/8/10)

#1 (1) bitNor
This one isn’t too bad.

#2 (2) bitXOR
Using NOT and AND, you can construct all logical operators. For instance,
X OR Y
is equivalent to
NOT (NOT X AND NOT Y)
(You should draw out a truth table to convince yourself.)

#3 (2) isNotEqual
Recall that the “truth” of something is defined as False if the value is 0 and True otherwise.
!0 = 1
!1 = 0
!(587) = 0

#4 (2) getByte
You can use shifts to isolate the “important” byte.

#5 (2) copyLSB
You can actually do this with just shift operations.

#6 (3) logicalShift
#7 (4) bitCount

#8 (4) bang
One way to solve this problem relies on a cute trick. If x=0, x is not negative. If you find –x, it is also
equal to 0, and is also not negative.

#9 listBitPos (4)
The only bit set in both x and –x will be the least significant one (check this on paper).

#10 (1) tmax
This one isn’t too bad.

#11 (3) isNonNegative
You’re going to want to focus on doing manipulations based on the most significant bit, since that’s
the sign bit.

#12 (3) isGreater
#13 (2) divpw2
#14 (4) abs

#15 (3) addOK
Adding a positive and a negative number is never a problem. Adding two positive numbers resulted
in an overflow if the answer is negative. Adding two negative numbers resulted in an overflow if the
answer is positive.

Before, all problems needed be solved. Now, 6, 7, 12, 13, and 14 are optional (read as extra credit).