***Homework 1***

*Problem 2.71*

1. The given wrong function would not work because it does not sign extend the bits required. It shifts the required word to the right most position and produces output with more significant bits being the same and no sign extension-taking place.
2. int xbyte(packed\_t word, int bytenum)

{

int leftEnd = word << ((3 – bytenum) << 3);

return leftEnd >> (3 << 3);

}

*Problem 2.82*

1. The output is 0. This expression is not true for the case when x= TMin and y=0. In this case, x<y would yield 1 while –x>-y would yield 0.
2. The output is 1. This is due to the ring properties of the 2’s compliment.
3. The output is 1. We know, ~x = -x-1 and ~y = -y-1 so ~x+~y+1 = -x-y-2+1 = -x-y-1 = ~(x+y) [~(x+y) = -(x+y) – 1]
4. The output is 1. This is because arithmetic operation of 2’s compliment and unsigned integers is the same.
5. The output is 1. Right shift divides the number and rounds it towards negative infinity so the number will always be either less than or equal to x.