Nathan Tung
CS33 Dis 1C
April 8, 2012

CS33 Homework #1

2.66)

```
int leftmost_one(unsigned x)
{
        x=(x>>1)|x;
        x=(x>>2)|x;
        x=(x>>4)|x;
        x=(x>>8)|x;
        x=(x>>16)|x;
        x=x-(x>>1);
        return x;
}
```

By shifting x to the right for each power of two less than its number of bits, then using the bit-or function each time, we ensure that everything after the first one becomes a one. Then by right-shifting one more time, we can subtract all the ones after it – leaving only the first one.

2.71) A. The function does not extend the sign for the extracted byte due to the type packed_t being unsigned. (Therefore, we can type-cast it into a signed integer.)

B. First, shift the word left based on the byte number. Then shifting the word to the right 24 times will yield the proper byte, to which sign extension is performed.

```
int xbyte(packed_t word, int bytenum)
{
        return (int) (word << ((3-bytenum)<<3)) >> 24;
}
```

2.72) A. The sizeof operator returns value of type size_t, which is defined as unsigned. Therefore the equation maxbytes-sizeof(val) results in an implicitly unsigned integer that is never less than zero, even if maxbytes is greater than the actual signed value of val's size.

B. The conditional test could include an explicit cast to a signed int on the sizeof function: if(maxbytes-**(int)**sizeof(val)>=0)

2.81) A. **No**, it does not always yield 1. Take x = TMin, y=-1, where x<y; but –x causes TMin to overflow to a 1, and –x>-y does not hold.

B. **Yes**, it always yields 1. This is true due to the ring properties which apply to two's complement arithmetic.

C. **Yes**, it always yield 1. We know that –x = ~x + 1, so ~x=-x-1.
Thus ~x+~y+1=-x-y-2+1=-x-y-1 and ~(x+y)=-(x+y)-1=-x-y-1, so the given equation holds.

D. **Yes**, it always yields 1. Both signed and two's complement addition work the same in terms of bit-level behavior.

E. **Yes**, it always yields 1. To perform a logical shift only to push it back would result in the same number. But an arithmetic right shift only approaches the value of negative infinity and gets smaller.