**1.**

Assume:

int x = rand();

int y = rand();

unsigned ux = (unsigned) x;

Are the following statements always true?

**a.**

ux >> 3 == ux/8

True, unsigned integers doesn’t need bias for right shift

**b.**

given x > 0,

((x << 5) >> 6) > 0

False, if x has 1 as its most significant bit, right shift will produce a negative number

**c.**

~x + x >= ux

True, since when comparing to unsigned integer, ~x + x (111…1) becomes UMax

**d.**

given x & 15 == 11,

( ~((x >> 3) & (x >> 2)) << 31) >= 0

False

1. Final comparison checks if the most significant bit is 0

2. From given condition we get x & 1111 == 1011, x= 1011

3. x >> 3 has least significant bit (lsb) of 1, x >> 2 has lsb of 0

4. & has lsb of 0, then negate has lsb of 1, left-shifting by 31 gets msb of 1, which is false

**e.**

given ((x < 0) && (x + x < 0))

x + ux < 0

True, since x is casted to unsigned

**f.**

given ((x < 0) && (y < 0) && (x + y > 0))

((x | y) >> 30) == -1

False

1. From given condition we get msb of x and y are both 10

2. Right shifting (x | y) is 111…10 (-2)

**2.**

Given: x has a 4 byte value of 255

What is the value of the byte with the lowest address in a

0x000000FF

**a.**

big endian system?

0x00

**b.**

little endian system?

0xFF