COMP.2030 LAB 10/11/23

1. In a 32-bit machine, consider two arbitrary values x and y:

int x = random(); unsigned ux = (unsigned) x;

int y = random(); unsigned uy = (unsigned) y;

For each of the following C expressions, indicate whether the expression always yields 1 (true). If it does not always yield 1, give an example that make it yield 0 (false).

* 1. ((x+y)<<4) + y -x == 17\*y+15\*x

True: even when overflow

* 1. ~x + ~y + 1 == ~(x+y) (Note: -x = ~x + 1 except when x=Tmin)

False: -x=~x+1, except when x=Tmin.

Therefore, if either x or y is Tmin, == returns 0 (false).

* 1. (ux-uy) == -(unsigned)(y-x)

True: Both sides are unsigned, and identical

* 1. (x < y) == (-x > -y)  
       
      False: when x=Tmin, -x=Tmin

* 1. ux – uy == -(y – x)  
       
      True

* 1. (x >= 0) || (x < ux)

False: for any x<0.  
(x<ux) is evaluated as unsigned and is always false (ux==ux).