Homework 3

1. In C language, if an evaluation expression contains both unsigned and signed values, then signed values will be implicitly casted into unsigned ones before evaluation. Please fill the following table with “<”, “>” or “=”. (Assume **int value is encoded using 16 bits**)

Constant A	Constant B	A ? B
-2U	-1U	
-1	1	
-1	100U	
-1	65535U	
-32767	32768U	

2. There is a illustration of code vulnerability similar to that found in FreeBSD’s implementation of getpeername(). Find one bug in the following codes and try to fix it.

```
/* Copy n bytes from src to dest */
/* Note: size_t means unsigned int */
void *memcpy(void *dest, void *src, size_t n);

/* Kernel memory region holding user-accessible data */
#define KSIZE 1024
char kbuf[KSIZE];

/* Copy at most maxlen bytes from kernel region to user buffer */
/* Must not copy more than maxlen bytes */
  int copy_from_kernel(void *user_dest, int maxlen) {
     /* Byte count len is minimum of buffer size and maxlen */
     int len = KSIZE < maxlen ? KSIZE : maxlen;
     memcpy(user_dest, kbuf, len);
     return len;
}
```

3. Assume x and y are both 4 bit signed integers. Fill the following table. Truncate all the results to 4 bits with 2’s complement and write their value in decimal.

	x+y	x-y	x*y	-y
x=4, y=7				
x=-6, y=-8				
x=5, y=-1				
x=-3, y=6				