1.) What is the 8 bit binary representation of the decimal number 45?
0010 1101

2.) What is the hexadecimal representation of the binary number 1010111?
0x57

3.) What is the decimal representation of the hexadecimal number 0x4a?
74

4.) What is the 8 bit two’s complement representation of the decimal number -110?
1001 0010

5.) Consider the following 12-bit two's complement representation of an integer x:
   \[ x = 1010 \ 1101 \ 0110 \]
   What is the decimal value of x?
   \[-2^{11} + 2^9 + 2^7 + 2^6 + 2^4 + 2^2 + 2^1 = -1322\]

6.) What would the following C expression evaluate to?
   ```c
   int val = (27 & 61) | ! (4 & 7);
   ```
   
   27 & 61 = 25
   4 & 7 = 4
   !4 = 0
   25 | 0 = 25

7.) What value will be stored in the variable val after this C expression executes?
   Express your answer in hexadecimal.
   ```c
   char val = ( 0xa4 >> 3 ) << 2 ;
   ```
   
   0xa4 = 10100100
   0xa4 >> 3 = 11110100. or 00010100
   0xa4 >> 3 << 2 = 11010000 or 01010000 (more correct)
   0xd0 or 0x50
   See in class discussion

8.) What is the value of result after the execution of this snippet of C code?
   ```c
   char c = -4;
   unsigned char u = 4;
   int result = c > u;
   ```
   
   It’s a mixed expression so both treated as unsigned but C spec doesn’t say how.
   ```gdb
   p ((unsigned char)-4) > ((unsigned char)4)
   $5 = 1
   p c > u
   $6 = 0
   ```
9.) For the following question, assume the primitive C type char is used to represent 8-bit two's complement numbers. What will the following C code print?

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
    int count = 0;
    char val = 0;

    while(val <= 0){
        val++;
        count++;
    }

    printf("%d", count);
}
```

128

This is because char is a signed int with 8 bit representation. When count is set to 128, val is also set to 128 but that is -128 in two’s complement notation, a negative value.

10.) How many unique values can be represented with 6 bits?

64