1.) What is the 8 bit binary representation of the decimal number 56?
   \[0011\ 1000\]

2.) What is the hexadecimal representation of the binary number 1001100?
   \[0x4c\]

3.) What is the decimal representation of the hexadecimal number 0xb2?
   \[178\]

4.) What is the 8 bit two’s complement representation of the decimal number -97?
   \[0110\ 0001 \Rightarrow 1001\ 1110 \Rightarrow 1001\ 1111\]

5.) Consider the following 12-bit two's complement representation of an integer \(x\):
   \[x = 1110\ 1001\ 0101\]
   What is the decimal value of \(x\)?
   \[-2^{11} + 2^{10} + 2^9 + 2^7 + 2^4 + 2^2 + 2^0 = -2048 + 1024 + 512 + 128 + 16 + 4 + 1 = -363\]

6.) What would the following C expression evaluate to? Specify your answer in decimal.
   \[
   \text{int val} = (31 \& 52) \mid ! (23 \& 8);
   \]
   \[
   31\&52 = 1\ 1111\ 110100 = 10100\ \\
   23 \& 8 = 1\ 0111\ 1000 = 0\ \\
   !0 = 1\ \\
   10100 \mid 1 = 10101 = 21
   \]

7.) What value will be stored in the variable val after these C expressions execute? Assume arithmetic right shifts are implemented. Express your answer in hexadecimal.
   \[
   \text{char val} = 0\text{xd3};
   \text{val} = (\text{val} \ll 3) \gg 2;
   \]
   \[
   0\text{xd3} = 11010011\ \\
   0\text{xd3} \ll 3 = 10011000\ \\
   (0\text{xd3} \ll 3) \gg 2 = 11100110\ \\
   0\text{xe6}
   \]

8.) What is the value of result after the execution of this snippet of C code?
   \[
   \text{int i} = -15;
   \text{unsigned int j} = 15;
   \text{unsigned int result} = i < j;
   \]
   \[\text{It's a mixed expression so both treated as unsigned but C spec doesn't say how.}\]
   \[
   (\text{gdb}) \ p ((\text{unsigned int})-15) < ((\text{unsigned int})15)
   \]
\$6 = 0 \$

9.) For the following question, assume the primitive C type short is used to represent 16-bit two's complement numbers. What will the following C code print?

```
#include <stdio.h>

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

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

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

```
32768
This is because short is a signed int with 16 bit representation. When count is set to 32768, val is also set to the same bit representation, namely 1000 0000 0000 0000. However, in a short’s 16 bit two’s complement representation, that bit sequence is interpreted as the negative value -32768.
```

10.) How many unique values can be represented with 11 bits?

```
2^{11} = 2048
```