Exercises:

Part I

1. Represent 305 with 10 bits
2. Represent -305 with 10 bits
3. Represent -197 with 9 bits
4. Represent 197 with 9 bits
5. Represent “0101010101” in hexadecimal
6. Represent “1100111101” in hexadecimal
7. Represent “11001100” as decimal (unsigned representation)
8. Represent “11001100” as decimal (signed representation)

Part II

1. Assume ***b*** is a variable of size 4 bytes and is stored in a byte addressable memory at address 0xA80. If the processor’s endianness is little-endian, and the processor writes the value 0xA155F0D3 in the variable ***b*** which bytes will be written to each memory address.

2. Assume ***x*** is a variable of type pointer that points to a single byte. Further, assume ***b*** is a variable of size 4 bytes and is stored at memory address 0xA80. Given that the processor uses big-endian, evaluate the new value of b after the following code is executed:

```
b=0x2F552;
x=0xA81;
b=b+*x;
```

3. Given a variable ***b*** which is assumed to have a value in the range [0..7], write the necessary statements in C to ensure that the bit at bit position ***b*** in another variable ***c*** is set to one.

4. Write a statement in C such that for a given variable ***b*** the bit at position 3 is set to 0, the bit at position 5 is set to 1, the bit at bit position 2 is inverted. Assume that the variable ***b*** is of size 1 byte.

Answers:

Part I

1. 305/2 = 152   1
   152/2 = 76   0
   76/2 = 38   0
   38/2 = 19   0
   19/2 = 9   1
   9/2 = 4   1
   4/2 = 2   0
   2/2 = 1   0
   1/2 = 0   1
   0/2 = 0   0   (0100110001)

2. ~0100110001=1011001110
   + 1
   =1011001111   (-305)

3. $-197=(-1)*2^8+X=-256+X$ → (1 _ _ _ _ _ _ _ _) X=59 with 8 bits

   59/2 = 29   1
   29/2 = 14   1
   14/2 = 7   0
   7/2 = 3   1
   3/2 = 1   1
   1/2 = 0   1
   0/2 = 0   0
   0/2 = 0   0   → (-197)= (100111011)

4. ~100111011=011000100

   $\underline{+ \qquad\qquad 1}$

   =011000101 (197)

5. 0101010101- unsigned 0001|0101|0101→ 0x155

   0101010101- signed 0001|0011|0011→ 0x155

6. 1100111101- unsigned 0011|0011|1101→ 0x33D

   1100111101- signed 1111|0011|1101→ 0xF3D

7. 11001100→ $1*2^7+1*2^6+0*2^5+0*2^4+1*2^3+1*2^2+0*2^1+0*2^0$ = 204

8. 11001100→$(-1)*2^7+1*2^6+0*2^5+0*2^4+1*2^3+1*2^2+0*2^1+0*2^0$= -52

Part II

1.

Address	Value
0xA80	0xD3
0xA81	0xF0
0xA82	0x55
0xA83	0xA1

2. First, the value 0x2F552 needs to be extended to 32 bits, i.e. ***b***=0x0002F552. This variable will be stored in memory as

Address	Value
0xA80	0x00
0xA81	0x02
0xA82	0xF5
0xA83	0x52

   As ***x*** points to memory address 0xA81, the expression ****x*** is evaluated as 0x02 (see table above).
   The new value of ***b*** is then

```
  0x0002F552
+ 0x00000002
  0x0002F554
```

3. ***c*=(1<<*b*)|*c*;**

4. ***b*=((*b* & 0b11110111) | 0b00100000) ^ 0b00000100;**