No calculators. Questions add to 104 points to allow for extra credit. You will be graded out of 100 points.

1. Convert these numbers between signed decimal and 16-bit two's complement representation.
   (5 pts. each)
   
   **signed decimal**  |  **two's complement**
   -------------------|---------------------
   a. +155             |                     
   b.                   | 0x0155              
   c. -155             |                     
   d.                   | 0xff55              

2. Give the most negative 12-bit two's complement number in two formats:  (4 pts. each)
   a. hexadecimal representation (two's complement): ____________
   b. decimal representation: ____________

3. Give the most positive 12-bit two's complement number in two formats:  (4 pts. each)
   a. hexadecimal representation (two's complement): ____________
   b. decimal representation: ____________

4. Show the hexadecimal results of binary addition and subtraction of signed 16-bit two's complement numbers. Identify any signed overflows. (5 pts. each)
   
   a. 0x7654
      +0x1abc
      -------
   b. 0x7654
      +0xffed
      -------

5. Show the hexadecimal results of logic operations on 16-bit binary numbers.  (4 pts. each)
   
   a. 0x7654 or 0xcedef
   b. 0x7654 and 0xcedef
   c. 0x7654 shift left by 1
   d. 0xcedef shift right logical by 2
   
   ---------
   ---------
   ---------
   ---------
6. Give a sequence of ARM shift and add instructions to multiply the value in register r0 by 20 decimal and place the result in register r1. You may use registers r2 and r3 as temporaries. (Do not use the mul instruction) (6 pts.)

9. Show the big-endian and little-endian byte ordering of the following data values. Start the byte numbering of t[] at address 100. (8 pts.)

```c
char t[4] = {'a', 'b', 'c', '0'};
short int u = 0x123; /* short int is 16-bit half word */
short int v = 0x45;
```

byte address: 100 101 102 103 104 105 106 107

big-endian ordering: ___ ___ ___ ___ ___ ___ ___ ___
little-endian ordering: ___ ___ ___ ___ ___ ___ ___ ___

10. Show the HARDWARE steps required in multiplication of two unsigned 4-bit binary numbers, 1110 times 1011. (1110 is the multiplicand and 1011 is the multiplier.) The details of mulsc and are not required, but you should show the 3-register format of ACC, MQ, and MDR, and be sure to place the multiplicand and multiplier in the correct registers. (12 pts.)
11. Show the ARM code for the following program segment. (12 pts.)

\[
d = d \ll 8; \\
\text{for( } i=0; i<8; i++ \text{ )}\{ \\
a = a \ll 1; \\
a = a - d; \\
\text{if( } a<0 \text{ )} \\
\quad \{ \\
\quad \quad a = a + d; \\
\quad \}
\text{else} \\
\quad \{ \\
\quad \quad a = a + 1; \\
\quad \}
\}
q = a \& 0xff; \\
r = (a>>8) \& 0xff;
\]

You may assume these register defines:

\[
a .req \ r0 \\
d .req \ r1 \\
a .req \ r2 \\
r .req \ r3 \\
\]

Add others as needed.

12. Explain what the code segment in question 11 does. (4 pts.)