UNCA CSCI 235
Final Exam Fall 2018
11 December 2018 – 3:00 pm to 5:30 pm

This is a closed book and closed notes exam. Communication with anyone other than the instructor is not allowed during the exam. **Furthermore, calculators, cell phones, and any other electronic or communication devices may not be used during this exam.** Anyone needing a break during the exam must leave their exam with the instructor. Cell phones or computers may not be used during breaks.

Name:_____

Problem 1 (10 points) C expressions
In the left column, there are fifteen tricky and not-so tricky C expressions. Write their values in the right column. Express your answers as simple base 10 expressions, such as 235 or -235. You may assume that all of these numbers are stored in 16-bit two's complement representation, the usual `short`.

`0123`	
`0xEB`	
`14 && 14`	
`42 & 35`	
`42 >> 3`	
`42 | 35`	
`42 << 3`	
`42 ^ 35`	
`~42 + 1`	
`3 * 4 / 5`	
`(3 * 4) / 5`	
`1 < (2 < 3)`	
`17 & ~17`	
`17 && -17`	
`(17 == 17) * 17`	

Problem 2 (4 points) Decimal to two's complement conversion
Convert the following four signed decimal numbers into **five**-bit *two's complement* representation. Some of these numbers may be outside the range of representation for **five**-bit two's complement numbers. Write "out-of-range" for those cases.

15	**32**
-15	**-32**

Problem 3 (3 points) Q4.4 to decimal conversion
Convert the following two Q4.4 *two's complement* numbers (four fixed and four fractional bits) into conventional decimal numbers.

11001000	**00010011**

Problem 4 (3 points) Decimal to Q4.4 conversion
Convert the following two signed decimal numbers into Q4.4 *two's complement* numbers (four fixed and four fractional bits). If you can't express the number exactly, give the nearest Q4.4 representation.

3.3	-1.414

Problem 5 (6 points) Adding numbers with flags
Add the following pairs of six-bit numbers. Based on the result of this addition, set the four x86-64 status bits: CF (carry), OF (overflow), SF (sign) and ZF (zero).

111011 **+ 000101** **CF__, OF__, SF__, ZF__**	**011100** **+ 000100** **CF__, OF__, SF__, ZF__**
100000 **+ 100000** **CF__, OF__, SF__, ZF__**	**010110** **+ 000110** **CF__, OF__, SF__, ZF__**

Problem 6 (2 points) Range
What is the range of numbers that can be stored in 16-bit twos-complement numbers? (The `short` of Java is a 16-bit twos-complement number.)

Problem 7 (2 points) Range
What is the range of numbers that can be stored in 16-bit unsigned numbers? (The `char` of Java is a 16-bit unsigned number.)

Problem 8 (6 points) Formatted printing

Suppose that the `int` variable C has the value 170 (in decimal). The left column in the table below has a `printf` statement. The right column has the desired output for that `printf` within a six character field. Your task is to fill in the underlined part (the stuff after the %). ***You must use a single "conversation specifier" (the thing starting with a %) in your format string. No "ordinary characters" are allowed.*** This means the following are not allowed because they contain ordinary characters.

```
printf("000160", C) ;   // contains only ordinary characters
printf("   %3d", C) ;   // starts with three ordinary characters
```

`printf("%_____",C) ;`	`_ _ _ 1 6 0`
`printf("%_____",C) ;`	`0 0 0 1 6 0`
`printf("%_____",C) ;`	`_ _ + 1 6 0`
`printf("%_____",C) ;`	`_ _ _ 2 4 0`
`printf("%_____",C) ;`	`_ _ _ _ a 0`
`printf("%_____",C) ;`	`_ _ _ _ A 0`