CS 22300 **Exam 1 Review Problems** Fall 2015

1. You should solve the following "Practice Problems" from the textbook (the solutions are on pages 143–161).

   Practice Problem 2.1 (page 37)

   Practice Problem 2.2 (page 37)

   Practice Problem 2.3 (page 38)

   Practice Problem 2.5 (page 48)

   Practice Problem 2.7 (page 49)

   Practice Problem 2.8 (page 51-52)

   Practice Problem 2.12 (page 55)

   Practice Problem 2.14 (page 57)

   Practice Problem 2.15 (page 57)

   Practice Problem 2.16 (page 58)

   Practice Problem 2.17 (page 65-66)

   Practice Problem 2.19 (page 71-72)

   Practice Problem 2.21 (page 76)

   Practice Problem 2.22 (page 79)

   Practice Problem 2.24 (page 82)

   Practice Problem 2.25 (page 83)

   Practice Problem 2.28 (page 89-90)

   Practice Problem 2.29 (page 93-94)

   Practice Problem 2.33 (page 95)

   Practice Problem 2.40 (page 103)

   Practice Problem 2.45 (page 111)

   Practice Problem 2.47 (page 117-118)

   Practice Problem 2.54 (page 125-126)

2. When `a` and `b` have the following binary values, fill in the resulting values for the operations in the following table. (Be sure to specify all 8 bits of each result.)

`a` = 01110010 `b` = 01010111

Operation	Value
`~a`	
`a & b`	
`a | b`	
`a ^ b`	
`b ^ b`	
`a || (b ^ b)`	
`a && ~a`	
`!a`	
`b && !b`	
`a << 4`	

3. Suppose that we have the following 8-bit binary value.

10011011

(a) What is the decimal value of this binary word if we interpret it as an unsigned 8-bit integer? Briefly explain your calculation.

(b) What is the decimal value of this binary word if we interpret it as a twos-complement 8-bit integer? Briefly explain your calculation.

4. What number does each line of code print out? Briefly explain why.

```
printf( "%u\n", (unsigned char)(-2) );
printf( "%d\n", (unsigned char)(-2) );

printf( "%u\n", (uint16_t)(-2) );
printf( "%d\n", (uint16_t)(-2) );

printf( "%u\n", (unsigned int)(-2) );
printf( "%d\n", (unsigned int)(-2) );
```

5. What number does each line of code print out? Briefly explain why.

```
printf("%d\n",  (0x40000000  << 1) >> 20);
printf("%d\n",  (0x40000000u << 1) >> 20);
```

6. Explain why `-1 < 0u` is false. That is, explain why the following code will print out `What?`.

```
if (-1 < 0u)
   printf("OK.\n");
else
   printf("What?\n");
```

7. Consider this code.

```
typedef union {
   int n;
   unsigned char bytes[4];
} MyUnion;

MyUnion u = { .bytes = { 0x57, 0x68, 0x6f, 0x00 } };

printf("%x\n", u.n );
printf("%d\n", u.n);
printf("%s\n", u.bytes);
```

   (a) What does this code print out on a big-endian computer?

   (b) What does this code print out on a little-endian computer?

8. Consider this code.

```
typedef union {
   int n;
   unsigned char bytes[4];
} MyUnion;

MyUnion u = { .bytes = { 0x57, 0x68, 0x6f, 0x00 } };

printf("%x\n", (short)u.n);
```

   (a) What does this code print out on a big-endian computer? Briefly explain why.

   (b) What does this code print out on a little-endian computer? Briefly explain why.

9. Fill in the missing information in the following table. Use fractional binary numbers, not floating point numbers.

Fractional Value	Binary Representation	Decimal Representation
		.25
5/8		
35/16		
	11.111	
	0.011	
		937.03125

10. Write the decimal number 937.03125 as a 32-bit, single precision IEEE floating point number. Write your result in both hexadecimal and in binary. Also, specify what the normalized significand is (before dropping the leading 1). And give the unbiased value of the exponent.

11. This question is about I/O redirection and pipes on the (cmd.exe) command line. Explain what each of the following possible command lines mean. In each problem, you need to associate an appropriate meaning to the symbols a, b and c (for example "a is the name of a program, b and c are the names of files" or "a and b are the names of programs and c is the name of a file" or "a is the name of a program, b and c are parameters to the program").

   (a) `z:\> a > b < c`

   (b) `z:\> a < b > c`

   (c) `z:\> a | b > c`

   (d) `z:\> a < b | c`

   (e) `z:\> a ; b ; c`

   (f) `z:\> a ; b > c`

   (g) `z:\> a ; b | c`

   (h) `z:\> a & b < c`

   (i) `z:\> a < b ; c`

   (j) `z:\> a < b & c`

   (k) `z:\> a & b | c`

   (l) `z:\> a &(b | c)`

   (m) `z:\> a b | c`

   (n) `z:\> a | b c`

   (o) `z:\> a & b ; c`

   (p) `z:\> a & b & c`

   See the following pages for the meanings of the special symbols `& < > | ;` .

   ```
   http://ss64.com/nt/syntax-redirection.html
   http://www.microsoft.com/resources/documentation/windows/
   xp/all/proddocs/en-us/ntcmds_shelloverview.mspx
   ```

12. What problem is there with the following two command lines?

   (a) `z:\> a | b < c`

   (b) `z:\> a > b | c`