CS270 Recitation 2
“Number Crunching”

This recitation will help you prepare for the quiz next section and for Homework Assignment 1.
The teaching assistant will go through examples of these problems, then you will do them.
Answers can be checked on the websites shown at the bottom of the assignment.

Goals

To understand data representation in a computer, including boolean, integer, floating point, and character values, and the associated logical and arithmetic operations.

Question 1 (10 points): What is the minimum number of bits required to represent the 54 fourteeners in the Colorado mountains? If you have exactly the minimum number of bits, how many bit patterns are unused?

Minimum number of bits: **6, since 2^6 = 64**

Number of unused bit patterns: **10, since 64 – 54 = 10**

Question 2 (10 points): What are the 8-bit binary and hexadecimal representations of the decimal value 45? Show all 8 bits.

Binary: **0b00101101**

Hexadecimal: **0x2D**

Question 3 (10 points): What is the range of unsigned integers that can be stored using 7 bits? What is the range for signed integers represented in 1’s and 2's complement, with the same number of bits?

Range of unsigned integers: **0** to **127**

Range of signed integers: **-63** to **63** (1's complement)

Range of signed integers: **-64** to **63** (2's complement)

Question 4 (10 points): Show the 2's complement addition of -32 plus 12, with both numbers in binary using 8 bits. Hint: make sure that the resulting binary number corresponds to the correct answer.

0b11100000 (-32) + **0b00001100** (12) = **0b11101100** (-20)

Question 5 (10 points): Show the 2's complement subtraction of 10 minus 8, with both numbers in binary using 8 bits. Hint: make sure that the resulting binary number corresponds to the correct answer.

0b00001010 (10) + **0b11111000** (-8) = **0b00000010** (2)

Question 6 (10 points): Show the results of the following bitwise operations (using the same number of bits as shown in each problem):

NOT(0b10101100) = **0b01010011**

0b10000010 OR 0b01110110 = **0b11110110**

0b10000111 AND 0b10111010 = **0b10000010**

0b10001000 XOR 0b01011111 = **0b11010111**

NOT(0b11011110 XOR 0b01100000) = **0b01000001**

Question 7 (10 points): Show the results of the following bitwise operations:

~(0x3456& 0xDCBA) = 0x**EBED**

(0xFF00 ^ 0x2244) | 0x1357= 0x**DF57**

Question 8 (10 points): Find the decimal floating-point numbers from the following values (assuming IEEE 32-bit floating-point representation):

0x417C0000 = **15.75f**

1 10000001 10100000000000000000000 = **-6.50f**

Question 9 (10 points): Find the binary and hexadecimal numbers for the following floating-point values (assuming IEEE 32-bit floating-point representation):

0.625f = **0x3F200000** (hexadecimal)

12.25f = **0b 0 10000010 10001000000000000000000** (binary)

Question 10 (10 points): Translate the following strings from characters into ASCII hexadecimal values and vice versa (show two hexadecimal digits per character):

"cs270" = **0x6373323730**

0x42696E617279 = "**Binary**"

Website for ASCII conversion: www.branah.com/ascii-converter
Website for IEEE floating-point conversion: http://www.h-schmidt.net/FloatConverter/IEEE754.html
Website for two's complement math: www.planetcalc.com/747