CSCI 1320 Computer Science I: Engineering Applications
Spring 2016
Instructor: Fleming
Assignment 10
Due Sunday, April 10th, by 11:55pm

**Decimal to binary to hex to ASCII conversion**

**Objectives**

- Convert numbers between different bases: decimal vs. binary vs. hexadecimal equivalents
- Extra credit

**What you need to do**

In class we talked about how to convert a decimal number to its binary equivalent. For this assignment, we would like you to think along the same lines to convert binary (base 2) numbers into their hexadecimal equivalents.

Your program should ask the user to input a positive integer between 0 and 255. Your program should then display, in a tabular fashion, the given number, and it's binary and hexadecimal equivalents.

**Task 1.** (**10 points)** Using the provided code ("dec2bin_arr.cpp" and "dec2bin_s.cpp"), write a function to convert any integer number between 0 and 255 to their binary equivalent. Please include the leading zeroes for the binary numbers.
**Task 2.** (**90 points)** Write a function that would convert an 8-bit binary number into it's hexadecimal equivalent. **Note:** if you find this option easier, it is ok to implement two functions, one for binary-to-decimal conversion, and one for decimal-to-hexadecimal conversion.

Sample run:
Please enter a number between 0 and 255:
**33**

| Decimal | Binary | Hexadecimal |
|---|---|---|
| 33 | 00100001 | 0x 21 |

You can see the equivalent conversions for integers 0-127 here:
http://upload.wikimedia.org/wikipedia/commons/d/dd/ASCII-Table.svg

**Task requirements**

Create a function for each of the conversions your solutions chooses to use: decimal to binary, decimal to hexadecimal, binary to decimal, binary to hexadecimal. You can

choose the type and number of the input argument(s) and the type of each of the return parameters to suit your programming solution.

**Extra Credit**
For extra credit, add another column containing the **Roman numeral** equivalent of the user input decimal number (extra **10 points**).

It has been a long time since you had to even THINK about Roman numerals vs. Arabic numerals, right? A Roman numeral looks like this "XVI" but an Arabic numeral looks like this "16." Here are the commonly used Roman numerals:

| | | |
|---|---|---|
| I = 1 | L = 50 | M = 1000 |
| V = 5 | C = 100 | |
| X = 10 | D = 500 | |

Here is an entire Roman Numeral table:

**Roman Numeral Table**

| | | | |
|---|---|---|---|
| **1** I | **14** XIV | **27** XXVII | **150** CL |
| **2** II | **15** XV | **28** XXVIII | **200** CC |
| **3** III | **16** XVI | **29** XXIX | **300** CCC |
| **4** IV | **17** XVII | **30** XXX | **400** CD |
| **5** V | **18** XVIII | **31** XXXI | **500** D |
| **6** VI | **19** XIX | **40** XL | **600** DC |
| **7** VII | **20** XX | **50** L | **700** DCC |
| **8** VIII | **21** XXI | **60** LX | **800** DCCC |
| **9** IX | **22** XXII | **70** LXX | **900** CM |
| **10** X | **23** XXIII | **80** LXXX | **1000** M |
| **11** XI | **24** XXIV | **90** XC | **1600** MDC |
| **12** XII | **25** XXV | **100** C | **1700** MDCC |
| **13** XIII | **26** XXVI | **101** CI | **1900** MCM |

Let's review some Roman numerals rules:

1) You cannot repeat a Roman numeral more than three times. In other words, if you want to write the Arabic number "30" as a Roman numeral, you can do it like this: XXX. But if you want to write the Arabic number "40" as a Roman numeral, XXXX would be incorrect. Instead, you would document XL. When you place a smaller Roman numeral in front of a larger Roman numeral, this indicates subtraction. So in our "XL" example, X=10 and L=50. And 50-10 = 40.

2) If smaller numerals follow larger ones, then you add. The same “no repeating more than three in a row” still applies. So if I want to express the number “11”, I write XI. For “12” I document XII. For “15” I write XV and so on.

**Submitting the assignment:**
Submit just the .cpp file containing your main() function. Make sure the program is well commented and it includes in the info header your name, course number, assignment number and instructor name. Submit the .cpp file through Moodle as Assignment 10 by Sunday, April 10th, by 11:55pm.