CS215: Introduction to Program Design, Abstraction and Problem Solving (Spring, 2023)

Programming Assignment 1

(100 points)

Today’s Date: Thursday, February 9

Due Date: Friday, February 24

Problem Statement:

The **Super Bowl** is among the world's most-watched single sporting events and frequently commands the largest audience among all American broadcasts during the year. Numerals **I** through **IV** were added later for the first four **Super Bowls**.

You will write a C++ program that repeatedly asks the user to input a year for the Super Bowl, and displays the corresponding Super Bowl with Roman Numeral, until the user enters either “Q” or “q” to quit your program.

You need to define the following three constants for your program:

```
// The first Super Bowl was hold in 1967 (at Los Angeles Memorial Coliseum)
const int START_SUPERBOWL = 1967;

// define the correct range for Roman Numerals: [MIN_ROMAN, MAX_ROMAN]
const int MIN_ROMAN = 1;
const int MAX_ROMAN = 3999;
```

You are given a Demonstration program during Today’s Lecture, which implements a “Roman Numeral Converter”, however it only works for one-digit or two-digit decimal numbers (at the range of [1, 99]).

Roman Numbers. (The following description is from ZyBook Programming Project 3.21.13). The Roman number system has digits

I	1
V	5
X	10
L	50
C	100
D	500
M	1,000

Numbers are formed according to the following rules:
(1) Only numbers up to 3, 999 are represented.
(2) As in the decimal system, the thousands, hundreds, tens, and ones are expressed separately.
(3) The numbers 1 to 9 are expressed as

I	1
II	2
III	3
IV	4
V	5
VI	6
VII	7
VIII	8
IX	9

As you can see, an I preceding a V or X is subtracted from the value, and you can never have more than three I's in a row.

(4) Tens and hundreds are done the same way, except that the letters X, L, C and C, D, M are used instead of I, V, X, respectively.

As part of the simpler smaller sub-problem to solve in the given demo program, it can only convert the number at the range of [1, 99] to its corresponding Roman numeral. For example, if you run this program with an input number, say 68, and it will convert **68** to Roman numeral, **LXVIII**.
In this project, you need to (1) ask the user to input a four-digit decimal number (representing a year); (2) if the input year is at the range of [1967, 1967+3999−1], your program should display the Super Bowl with the Roman Numeral representation: **1967** is Super Bowl **I**; **1968** is Super Bowl **II**; **1969** is Super Bowl **III**; **1970** is Super Bowl **IV**…; **2022** is Super Bowl **LVI** ; **2023** is Super Bowl **LVII** ... and so on.
Based on the source code of Demonstration program, you should modify the solution to a smaller problem into the complete solution to the bigger problem in this Project. You also need to follow the requirements shown below.

Special Requirements:

1. You are required to define the following function in Project 1, named **`roman_numeral()`**, so that **`main`** function can call this function to solve the problem:

```
/*
   Purpose: convert the integer n to its corresponding Roman Numeral
            n must be between 1 and 3999
            it returns string form of the Roman Numeral
   @param n int: representing the number to convert into Roman Numeral
   @return string: representing the corresponding Roman Numeral for n
*/
string roman_numeral(int n);
```

(Please note that you can define other functions if you think you need to reuse the same block of code more than twice.)

2. Design user-friendly interface: your program should print out the follow information when collecting the user input:

```
***********************************************************
*     The Super Bowl is the annual final playoff game     *
*         of the NFL to determine the league champion.    *
* The first Super Bowl took place on January 15, 1967.    *
* Super Bowl I (Los Angeles Memorial Coliseum) --> 1967   *
* This Roman Numerals Convertor is written by Yi Pike.    *
* If you had a time machine, which year of Super Bowl     *
* you want to attend (1967 - 5965) ?                      *
***********************************************************
Please enter the year you want to attend (click Q or q to quit):
```

However, you need to use your name to replace the instructor's name shown above. Why the range of the years is from 1967 to 5965? Because the first Super Bowl took place in 1967, and the largest valid Roman number is 3999: starting from 1967, the largest year which can be represented using Roman number is 1967 + 3999 – 1 = 5965. Think about how to avoid "Magic Number" in your program!

3. User input validation is required for this Project, which means that you can no longer assume that the user will always input "Good Data".

After passing the compilation, please download the following sample output file to test running your program, and think about how to design your own testing cases:

http://www.cs.uky.edu/~yipike/CS215/Sample_SuperBowl.pdf

Submission:

Open the link to Course Canvas page (https://www.uky.edu/canvas), and login to your account using your LinkBlue ID and password. Please submit your source code in a .cpp file through link "**`Project 1`**". It is a good idea to check that your file is already uploaded successfully. If not, go back and submit it again.

(Late assignment will be reduced 10% for each day that is late. The assignment will not be graded (you will receive zero) if it is more than 5 days late. Note that a weekend counts just as regular days. For example, if an assignment is due Friday and is turned in Monday, it is 3 days late.)

Please check the Grading sheet on next page for your reference.

Grading Sheet for Programming Assignment 1

Total: 100 points.

	Points	Deducted Points
Correctness	55	
Your program repeatedly asks the user to enter a year until the user enters either “Q” or “q” to quit the program.	5	
Your program handles User Input Validation correctly: (instead of quitting the program)		
• Display the message of invalid input and let the user try again	5	
• Display the corresponding message when the user input is earlier than 1967, and let the user try again	5	
• Display the corresponding message when the user input is later than 5965, and let the user try again	5	
Your program converts the user-input year into the corresponding Roman Numerals correctly	22	
Your program defines constants required by the problem statement. (you can define other constants if you need)	3	
Your program defines the function, named roman_numeral(), so that main() function calls this function to solve the problem. (if your program solves the problem, however, did not define this function to call in the main, you lose 10 points)	10	
Style	15	
Lay out your program in a readable fashion	5	
Include comments as specified in the lecture notes	5	
User-friendliness in I/O design	5	
Testing (No Documentation is required)	30	
Pass testing cases of valid user input, such as:	18	
Case 1, the user types 1967;		
Case 2, the user types a year in 1970s, 1980s, 1990s, 2000s, 2010s, 2020s, and so on;		
Case 3, the user types the year of 3998, 3999, 4000 and 4001;		
Case 4, the user types the year of 5964, 5965, 5966 and 5967.		
Pass testing cases of invalid user input, such as:	12	
Case 1, the user types an integer which is not in the right range:		
1.a: a negative number, such as -1978 or -5;		
1.b: an integer smaller than 1967;		
1.c: an integer larger than 5965.		
Case 2, the user types a few string when an integer is being expected, such as “Two Thousand and Two”		
Case 3, the user types a double floating point number when an integer is being expected, such as “2022.58 ok?”		
Case 4, the user types an integer with letters before and after, when an integer is being expected, such as “do you think 2022 is ok?”		
Your Score		