

**CS215: Introduction to Program Design, Abstraction and Problem Solving  
(Fall, 2015)**

**Programming Assignment 1  
(100 points)**

Today's Date: Thursday, September 10

**Due Date: Sunday, September 20**

**No late submission will be allowed!**

**Problem Statement**

Write a program that allows the user to repeatedly enter positive integers to be converted into the Roman number system until the user enters -1 to exit the program.

The Roman number system has digits:

<b>M</b>	<b>1000</b>
<b>D</b>	<b>500</b>
<b>C</b>	<b>100</b>
<b>L</b>	<b>50</b>
<b>X</b>	<b>10</b>
<b>V</b>	<b>5</b>
<b>I</b>	<b>1</b>

Numbers are formed according to the following rules:

- Only numbers up to 3999 are represented.
- As in the decimal number system, the thousands, hundreds, tens, and ones are expressed separately.
- The numbers 1 to 9 are expressed as

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

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.

- 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.

A sample output of your program can be as follows:

**Integer--Roman Number conversion Tool:**

**Enter -1 to quite the program.**

**Please enter a positive integer less than 4,000 you wish to convert: 2015**

**The roman numeral is MMXV**

**Integer--Roman Number conversion Tool:**

**Enter -1 to quite the program.**

**Please enter a positive integer less than 4,000 you wish to convert: 1978**

**The roman numeral is MCMLXXVIII**

**Integer--Roman Number conversion Tool:**

**Enter -1 to quite the program.**

**Please enter a positive integer less than 4,000 you wish to convert: 215**

**The roman numeral is CCXV**

**Integer--Roman Number conversion Tool:**

**Enter -1 to quite the program.**

**Please enter a positive integer less than 4,000 you wish to convert: -215**

**Your number must be greater than zero and less than 4000**

**Integer--Roman Number conversion Tool:**

**Enter -1 to quite the program.**

**Please enter a positive integer less than 4,000 you wish to convert: 1865**

**The roman numeral is MDCCCLXV**

**Integer--Roman Number conversion Tool:**

**Enter -1 to quite the program.**

**Please enter a positive integer less than 4,000 you wish to convert: 4008**

**Your number must be greater than zero and less than 4000**

**Integer--Roman Number conversion Tool:**

**Enter -1 to quite the program.**

**Please enter a positive integer less than 4,000 you wish to convert: 999**

**The roman numeral is CMXCIX**

**Integer--Roman Number conversion Tool:**

**Enter -1 to quite the program.**

**Please enter a positive integer less than 4,000 you wish to convert: 79**

**The roman numeral is LXXIX**

**Integer--Roman Number conversion Tool:**

**Enter -1 to quite the program.**

**Please enter a positive integer less than 4,000 you wish to convert: -1**

**Thank you for using Integer--Roman Number conversion Tool.**

In order to keep your program's window from closing, you can call `system("pause");` before returning from main. Your program must compile in order to be graded.

**Submission:**

Open the link to csportal (<http://www.cs.uky.edu/csportal>), and login to your account using your linkblue user id and password. Please zip the inside folder (project), and submit (upload) the zip file through link “PA1”. Note that only one file is allowed to upload and it should be your zip file. It is a good idea to check that your file is already uploaded successfully. If not, go back and submit again.

### Grading Sheet for Programming Assignment 1

Total: 100 points.

	Points	Deducted Points
Correctness	65	
Roman numbers are correctly generated for thousands digit, hundreds digit, tens digit and units digit.	12*4	
User is allowed to repeatedly use the program till enter -1 to quit.	6	
Program checks whether user input number is in the correct range.	6	
Submit your project as one zip file through CSPortal	5	
Style	20	
Lay out your program in a readable fashion	5	
Include comments as specified in the lecture notes	10	
User-friendliness in I/O design	5	
Testing	15	
Testing simple cases such as numbers shown in the table on page 1	5	
Testing normal cases such as examples in the sample output	5	
Testing invalid input	5	
Your Score		