Homework

09

Roman Times

Convert between Arabic, base ten numbers, and Roman numerals. The Roman number system has the digits **I**, **V**, **X**, **L**, **C**, **D**, and **M**.

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. (In other words, it's a positional number system.)

The numbers **1** to **9** are expressed like the table shown at the right. As you can see, a **I** preceding a **V** or **X** is subtracted from the value, and you can never have more than three **I** 's in a row.

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

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.

The **toRoman()** function accepts a decimal number such as **1978**, and converts it to a **string** containing the Roman numerals "**MCMLXXVIII**". An invalid number (**0**, a negative number, or a number greater than **3,999**) returns the **string** "**OUT OF RANGE**".

Do make things easier for you, I've already written a regular console program that converts integers to Roman numerals. You'll find it in the file **decToRoman.txt**. Notice that the program is long and complex. Decompose **toRoman()** using the following four functions, which will be tested separately as well.

```
string digit(int n, const string symbols);
string ones(int n) { return digit(n, "IVX"); }
string tens(int n) { return digit(n, "XLC"); }
string hundreds(int n) { return digit(n, "CDM"); }
```

Use **make test** to test your code, **make stest** or **make run** to run any student tests. Once your score is OK, use **make submit** to turn it in. If you get stuck, ask for help on Piazza, or come by my office hours (early!!!).