

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.

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.

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

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"**. Here is the pseudocode for the function.

```
ToRoman <- number
  if number is out of range return "OUT OF RANGE";
  onestr <- ones(number mod 10);
  number <- number / 10
  tenstr <- tens(number mod 10);
  number <- number / 10
  hundredstr <- hundreds(number mod 10);
  number <- number / 10
  thousands <- number * "M"
  return thousands + hundredstr + tenstr + onestr
```

You will need to implement both the **toRoman()** and the **digit()** functions. The **ones()**, **tens()** and **hundreds()** functions are **already implemented** as inline functions, inside the header file.

```
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"); }
```

The **digit()** function should translate **one digit**, using the **symbols** parameter specified for the one, five and ten values. You will need to break the **symbols** parameter into three parts: **symbols.at(0)** will be the ones character, **symbols.at(1)** will be the fives character, while **symbols.at(2)** will be the tens character.

Use **make test** to test your code 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 Canvas, or come by my office hours (early!!!).