

罗马数字和整数互相转换

罗马数字由7个不同的符号组成：I, V, X, L, C, D 和 M。

符号	值
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

举个栗子，2 用罗马数字写作 II，由两个 I 组成；12 写作 XII，由 X 和 II 组成。27 写作 XXVII，由 XX、V 和 II 组成。

罗马数字通常按值从左到右，由大到小组成。但是，4 并不是 IIII，而是写作 IV。因为1 在5之前，所以我们从5 减去1得到4。同样的原则适用于数字9，它被写作IX。总共有 6 种需要用到减法的情况：

- I 能够放在V(5) 和X(10) 之前，组成4和9。
- X 能够放在L(50) 和C(100) 之前，组成40和90。
- C 能够放在D(500) 和M(1000) 之前，组成400和900。

输入 (`input.txt`)

```
III
IV
IX
LVIII
MCMXCIV
64
428
3999
```

LVIII: $L = 50$, $V = 5$, $III = 3$

MCMXCIV: $M = 1000$, $CM = 900$, $XC = 90$, $IV = 4$

输出 (`output.txt`)

```
3
4
9
58
1994
LXIV
CDXXVIII
MMMCMXCIX
```

说明

- 本题主要考察
 - 基本的代码编写能力，需要遵循一定的类、方法、变量命名规范
 - 基本的源码组织能力，需要遵循一定的目录结构及文件夹、文件命名规范
 - 具体的转换算法可以参考网上公开的实现
- 请确保能够使用命令行或 IDE 进行构建及运行
 - 前端：Webstorm，VS Code 等工具
 - 后端：IDEA，Eclipse 等工具
- 请不要使用外部库来实现算法，可以使用基础库。如 JDK、Node等

- 请通过文本文件加载 `input.txt`，并确保结果同 `output.txt` 中给定的一致
 - 请自行创建 `input.txt`，生成 `output.txt`
 - 前端：请使用 [Node](#) 的文件处理 API 实现文件的读写
 - 后端：请使用 Java 的文件处理 API 实现文件的读写
- 公平起见，请不要分享或发布作业给第三方
- 完成后请将完整的项目打包成 Zip 文件，附加到回复邮件

NodeJS 读写文件示例

```
const fs = require('fs');  
// 读取文件  
const input = fs.readFileSync('input.txt', {encoding: 'utf8'});  
// 计算结果  
const output = calculate(input);  
// 写入文件  
fs.writeFileSync('output.txt', output, {encoding: 'utf8'});
```