

罗马数字转整数

罗马数字包含以下七种字符: **I**, **V**, **X**, **L**, **C**, **D** 和 **M**。

字符	数值
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

例如，罗马数字 **2** 写做 **II**，即为两个并列的 **1**。**12** 写做 **XII**，即为 **X** + **II**。**27** 写做 **XXVII**，即为 **XX** + **V** + **II**。

通常情况下，罗马数字中小的数字在大的数字的右边。但也存在特例，例如 **4** 不写做 **IIII**，而是 **IV**。数字 **1** 在数字 **5** 的左边，所表示的数等于大数 **5** 减小数 **1** 得到的数值 **4**。同样地，数字 **9** 表示为 **IX**。这个特殊的规则只适用于以下六种情况：

- **I** 可以放在 **V** (**5**) 和 **X** (**10**) 的左边，来表示 **4** 和 **9**。
- **X** 可以放在 **L** (**50**) 和 **C** (**100**) 的左边，来表示 **40** 和 **90**。
- **C** 可以放在 **D** (**500**) 和 **M** (**1000**) 的左边，来表示 **400** 和 **900**。

给定一个罗马数字，将其转换成整数。输入确保在 **1** 到 **3999** 的范围内。

示例 1:

输入: "III"

输出: 3

示例 2:

输入: "IV"

输出: 4

示例 3:

输入: "IX"

输出: 9

示例 4:

输入: "LVIII"

输出: 58

解释: L = 50, V = 5, III = 3.

示例 5:

输入: "MCMXCIV"

输出: 1994

解释: M = 1000, CM = 900, XC = 90, IV = 4.

解题思路: 根据示例 5 进行分析:

MCMXCIV

读到第一个字符为 M, 后一个为 C M>C 所以直接相加, 初始化 sum=0, sum=sum+s[0]=1000
当字符读到第二个字符 C, 后一个为 M C<M, 后面的字符大, sum=sum-s[1]=1000-100=900
当字符读到第三个字符 M, 后一个为 X X<M, 后面的字符小, sum=sum+s[2]=900+1000=1900
当字符读到第四个字符 X, 后一个为 C C>X, 后面的字符大, sum=sum-s[3]=1900-10=1890
当字符读到第五个字符 C, 后一个为 I C>I, 后面的字符小, sum=sum+s[4]=1890+100=1990
当字符读到第六个字符 I, 后一个为 V V>I, 后面的字符大, sum=sum-s[5]=1990-1=1989
当字符读到第七个字符 V, 后一个为 null 为空直接相加, sum=sum+s[6]=1989+5=1994

```
class Solution {
public:
    int romanToInt(string s) {
        //创建 hash 表
        map<char, int> m = {{ 'I', 1 }, { 'V', 5 }, { 'X', 10 }, { 'L', 50 }, { 'C', 100 }, { 'D', 500 }, { 'M', 1000 }};
        int tmp, sum = 0;
        //理解罗马数字的组合时看成左右两部分, 从左往右扫一遍 s,
        //如果遇到右边比左边大的, 则左边的这一部分便是负的, 否则直接加到底就好
        for (int i = 0; s[i] != '\0'; i++)
        {
            tmp = m[s[i]];
            if (s[i+1] && m[s[i+1]] > m[s[i]])
            {
                sum = sum - tmp;
            }
            else
                sum = sum + tmp;
        }
        return sum;
    }
};
```

```
    }  
};
```