

a013 羅馬數字

丁培毅

1

說明

- 如果生活在數世紀之前的古羅馬，應該會用 V 來表示五。V 和 5 這兩個符號都可以用來表示數目五。用來表示數目的符號稱作數字，而羅馬人用來表示數目的符號就是羅馬數字。
- 以下是七個基本的羅馬數字：

| 羅馬數字 | I | V | X | L | C | D | M |
|------|---|---|----|----|-----|-----|------|
| 數字 | 1 | 5 | 10 | 50 | 100 | 500 | 1000 |

- 所有其它的數目都是由這些數字組合而成。數目都是由左寫到右，通常值是等於組成的羅馬數字加起來：例如 17 可以表示為 $X+V+I+I=XVII$ 亦即 $10+5+1+1=17$
- 表示羅馬數字可以使用減法來取代加法的規則。例如四可以不用四個一相加來表示 IIII，而採用五減一來表示 IV。利用這類規則，羅馬人能夠減化許多數目的表示方式，例如使用 IX 取代 VIII 表示 9，使用 CD 取代 CCCC 表示 400 ----
連續四個相同符號出現時，必須用減法規則來化簡

2

題目

- **題目:** 讀入兩個正整數，輸出兩個數字差的絕對值。所有的數字都必須以羅馬數字來表示。而連續四個相同符號出現時，必須用減法規則來化簡之。
- **輸入說明:**
每個輸入檔中會有一個或以上的測試資料。每一行由兩個數字組成一筆測試資料，且所有數字將會小於 4000。檔案最後會以符號 # 表示結束。
- **輸出說明:**
每筆測試資料的答案必須輸出到檔案中，並且換行。如果答案為零，則須輸出字串 ZERO。
- **範例輸入:**
I I
MM II
#
- **範例輸出:**
ZERO
MCMXCVIII

3

解題步驟 (1/6)

1. **瞭解題目:** 看到一個題目時第一件事情當然是清楚地了解題目的要求, 這常常可以從範例輸入和範例輸出快速地掌握 (不過題目的文字還是要仔細看過的)
... 題目要求做兩個羅馬數字的差的絕對值
... 兩個羅馬數字都在 1 到 3999 之間
... 範例輸入 I I, 輸出 ZERO (1減1為0)
... 範例輸入 MM II, 輸出 MCMXCVIII (2000 減 2 為 1998)
當然如果輸入 II MM, 輸出還是 MCMXCVIII
2. **分析:** 兩個羅馬數字直接比較大小或是計算差值似乎沒有特別簡單的作法, 如果把羅馬數字轉換成十進位數字, 至少比較大小和計算差值都很容易, 算完以後再換回羅馬數字表示, 所以如果我們可以寫出“羅馬數字=>十進位”以及“十進位 => 羅馬數字”的兩個轉換函式, 這個題目應該就可以很快完成

4

解題步驟 (2/6)

3. “十進位數字=>羅馬數字”，準備好基本範例，推導規律化的轉換規則

1, 2, 3 => I, II, III 10, 20, 30 => X, XX, XXX 100, 200, 300 => C, CC, CCC

1000, 2000, 3000 => M, MM, MMM

5, 6, 7, 8 => V, VI, VII, VIII 50, 60, 70, 80 => L, LX, LXX, LXXX

500, 600, 700, 800 => D, DC, DCC, DCCC

4, 9 => IV, IX 40, 90 => XL, XC 400, 900 => CD, CM

459 => 400 + 50 + 9 => CD L IX

2974 => 2000 + 900 + 70 + 4 => MM CMLXXIV

分析：由以上範例，可以看到羅馬數字其實是十進位制，由高位數到低位數，每一位數個別轉換為對應的一串字元符號，基本上符號出現的順序是相同 (II, XXX, CC, MMM) 或是由大到小 (MLX, VII, LX)，唯一例外是 4xx 和 9xx 會出現一個小的符號在前面 (IV, XL, CD)

轉換規則：需要運用陣列建立轉換表格，配合迴圈取得十進位每一位數，以條件判斷敘述來輸出，例如: 2974 => 2, 9, 7, 4

2000 => MM, 900 => CM, 70 => LXX, 4 => IV

5

解題步驟 (3/6)

4. “十進位數字 => 羅馬數字” 程式實作：

資料設計：

十進位數字 int number=2974, digits[4]; // 4 位數, digits[3] 為個位數

羅馬數字 char roman[17]=""; // 4*4+1 每一位數換成羅馬數字最多 4 字元

```
以迴圈取得十進位每一位數字 for (i=0; i<4; i++) {
                                digits[3-i] = number % 10;
                                number /= 10;
                                }
```

```
char *convTable[4][10] = {{"", "M", "MM", "MMM"},
                          {"", "C", "CC", "CCC", "CD", "D", "DC", "DCC", "DCCC", "CM"},
                          {"", "X", "XX", "XXX", "XL", "L", "LX", "LXX", "LXXX", "XC"},
                          {"", "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX"};
```

以迴圈串接轉換過的羅馬數字

```
for (i=0; i<4; i++)
    strcat(roman, convTable[i][digits[i]]);
```

撰寫函式 decimalToRoman(int number, char roman[])

6

解題步驟 (4/6)

4. 程式所使用的表格的大小，常常和程式碼的多寡成反比，觀察上面程式裡所使用的表格，可以看到其實很多字元是重複的，在個位數、十位數、百位數、千位數羅馬數字重複的方法是一致的，所以也可以用程式邏輯來使得表格變小一些，例如：

用到的羅馬數字符號 char rDigits[4][2] = {'M'}, {'C', 'D'}, {'X', 'L'}, {'I', 'V'};

```
十進位數字 => 羅馬數字
for (i=0, iroman=0; i<4; i++) {
    if (digits[i]<=3)
        for (j=0; j<digits[i]; j++) roman[iroman++] = rDigits[i][0];
    else if (digits[i]==4)
        roman[iroman++] = rDigits[i][0], roman[iroman++] = rDigits[i][1];
    else if (digits[i]<=8) {
        roman[iroman++] = rDigits[i][1];
        for (j=0; j<digits[i]-5; j++) roman[iroman++] = rDigits[i][0];
    }
    else if (digits[i]==9)
        roman[iroman++] = rDigits[i][0], roman[iroman++] = rDigits[i-1][0];
}
roman[iroman] = 0;
```

實際上也可以看成是用程式來產生表格

7

解題步驟 (5/6)

設計一個直覺的規律作法，手動根據這個作法來完成題目的需求

5. “羅馬數字=>十進位數字”，準備好基本範例，推導轉換規則

II => +1+1 => 2

IV => -1+5 => -1+5 => 4

VII => +5+1+1 => 7

XIV => +10-1+5 => 14

XVIII => +10+5+1+1+1 => 18

XIX => +10-1+10 => 19

XXXIX => 10+10+10-1+10 => 39

XL => -10+50 => 40

XLI => -10+50+1 => 41

MCMXC VIII => 1000-100+1000-10+100+5+1+1+1 = 1998

規則：基本上羅馬數字 $r_1 r_2 r_3$ 是由大到小 $r_1 \geq r_2 \geq r_3$ ，則 $d=r_1+r_2+r_3$

允許有一個數字順序顛倒，例如 $r_1 \geq r_2 < r_3$ 且 $r_1 \geq r_3$ ，則 $d=r_1-r_2+r_3$

轉換方法：

一個一個數字 r_1, r_2, r_3 轉為 10 進位數字，但是正負號由下一數字決定，例如若 $r_2 \geq r_3$ ，則 r_2 為正，否則若 $r_2 < r_3$ ，則 r_2 為負，如果出現 $r_1 < r_3$ 且 $r_2 < r_3$ 的情況就代表是不合法的羅馬數字

問題：如果輸入不合法的羅馬數字 IIV, XIIX, VX, IXL, XII, IVI, XLXX

程式不應該轉換成十進位數字，尤其是最後兩個，在上面的規則裡面沒有偵測出來，因為羅馬數字基本上是十進位制，所以十位數 XL 出現一次此後不會再出現第二次十位數 XX

8

解題步驟 (6/6)

6. 前面的方法如果羅馬數字是合法，就可以正確地轉換成十進位，可以另外加上一個檢查的函式 `int isLegal(char roman[])` 來檢查是否為合法的羅馬數字，仔細觀察一下，可以發現羅馬數字千位數一定是 **M** 起頭，百位數字一定是 **C** 或是 **D** 起頭，十位數字一定是 **X** 或是 **L** 起頭，個位數字一定是 **I** 或是 **V** 起頭，因此一段羅馬數字符號可以切割為幾個部份，例如 `MCMXC VIII => M, CM, XC, VII` 然後再透過前面步驟 3 裡面的 `convTable[4][10]`，以迴圈來比對是否為合法的編碼 (同時也可以得到對應的十進位數值)

上面過程中也許你覺得很繁瑣、很想跳過，別人早就有分析好的方法，何必自己浪費時間重新發明車輪，不是毫無創意嗎? 不過在這裡你的重點在於練習分析問題、以程序式的規律方法解決問題、並且轉換為語法還不太熟悉的 C 程式，一步一步練習你才能體會其中的細節，如果找一個目前還沒有解答的問題來練習，就有點難確定目標為何了?