

# 程式設計 (105-2)

## 作業九

作業設計：孔令傑  
國立臺灣大學資訊管理學系

繳交作業時，請至 PDOGS (<http://pdogs.ntu.im/judge/>) 為第一、二題各上傳一份 C++ 原始碼 (以複製貼上原始碼的方式上傳)、對第三題做同儕互評，再為第四題上傳一個 PDF 檔。第四題是 bonus 加分題。每位學生都要上傳自己寫的解答。不接受紙本繳交；不接受遲交。請以英文或中文作答。

這份作業的截止時間是 **2017 年 5 月 15 日凌晨一點**。在你開始前，請閱讀課本的第 9-11 章<sup>1</sup>。為這份作業設計測試資料並且提供解答的助教是李昱賢 (Rick Lee)。

### 第一題

(40 分) 在電腦的世界裡，非整數 (浮點數) 總是令人困擾，畢竟二進位的電腦在儲存非整數時常有精確度的損失。如果是有理數 (rational number，可以被表示成兩個整數相除的數)，我們有機會解決這個問題：不要用 float 或 double 來儲存有理數，而自己寫一個 class 來儲存有理數！只要我們在 class 裡面放兩個 instance variable 分別存放分子和分母，而不是直接 (不精確地) 儲存他們相除後的結果，就可以解決這個問題了<sup>2</sup>。

現在想像一下你跟你的夥伴要一起寫一個程式，其中會遇到許多有理數，而且你們經常要對這些有理數做運算和比較<sup>3</sup>。你的夥伴建議你寫一個 class 來儲存有理數。由於你從未寫過 class，你的夥伴設計了如下的架構：

```
class RationalNumber
{
private:
    int num; // the numerator
    int deno; // the denominator
public:
    RationalNumber(); // initialize a rational number to 0
    RationalNumber(int num, int deno);
        // initialize a rational number to num2 / deno2
    void print(); // print out num2 / deno2 as a quotient
    int getInteger(); // return the integer part of this number
    bool isLargerThan(RationalNumber rn);
        // return true if the number is larger than rn
```

<sup>1</sup>課本是 Deitel and Deitel 著的 *C++ How to Program: Late Objects Version* 第七版。

<sup>2</sup>這當然有個前提，就是這些分子跟分母都不超過整數的儲存範圍。在本題中我們假設這對於你的應用是成立的。

<sup>3</sup>這種應用當然很多。比如說你若開了數家零售店，並且記錄每家店每個月的交易次數和總營業額，則總營業額除以交易次數就是這家店的客單價。如果你要幫這些店用客單價排序，就需要做有理數的比較，此時記錄兩對分子分母 (共四個整數) 就比記錄兩個浮點數來得精確。當然，如果你處理商管、經濟領域的問題，上述的精確度問題通常就不太重要；但如果你處理科學或工程領域的問題，這個精確度就非常重要了。

```
};
```

他對你詳細地說明了這些 `member` 的意義：

- `num` 和 `deno`：分別儲存分子和分母。這兩個數字應該要隨時保持互質狀態（也就是兩者的最大公因數為 1）。
- `RationalNumber()`：把分子和分母分別設成 0 和 1。
- `RationalNumber(int num, int deno)`：如果 `num` 和 `deno` 互質，就把分子和分母分別設成 `num` 和 `deno`；如果不互質，就先進行約分再儲存互質的分子跟分母。
- `print()`：以分數而非小數的形式印出這個有理數，先印出 `num`，接著印出 `/`，最後印出 `deno`，中間不要空白。
- `getInteger()`：回傳這個有理數的整數部份，也就是 `num` 除以 `deno` 後去掉餘數的部份。
- `isLargerThan(RationalNumber rn)`：如果自己比 `rn` 大則回傳 `true`，反之則回傳 `false`。

請實作上面描述的 `class RationalNumber`，以完成這些要求的功能。當然，你可以（事實上是需要）加入自己想要的 `member`，但上述這些是必須遵照規格去完成的。

## 輸入輸出格式

系統會提供許多筆測試資料，每筆測試資料裝在一個檔案裡。一個檔案的每一列的第一個字元標示著這一系列的任務，後面接著相對應的參數：

- 如果一列的第一個字元是 `N`，表示要產生一個有理數。後面會接三個參數，第一個是長度不超過十個字元的此有理數的變數名稱，第二個是分子的值，第三個是分母的值。分子跟分母不一定互質。如果變數名稱和之前產生過的有理數重複，就用這一個新的有理數的值取代舊的。
- 如果一列的第一個字元是 `P`，表示要印出一個有理數。後面會接一個參數，是一個有理數的變數名稱。如果確實有一個變數是這個名稱，就印出該有理數的分數型態，反之則什麼都不要印出（或者印出空字串）。
- 如果一列的第一個字元是 `I`，表示要印出一個有理數的整數部份。後面會接一個參數，是一個有理數的變數名稱。如果確實有一個變數是這個名稱，就印出該有理數的整數部份，反之則什麼都不要印出。
- 如果一列的第一個字元是 `C`，表示要比較兩個有理數。後面會接兩個參數，各是一個變數名稱。如果確實有兩個變數是這兩個名稱，就做比較，若第一個比第二個大，就印出 `larger`，若相等則印出 `equal`，若比較小則印出 `smaller`。如果任何一個名稱找不到相對應的變數，就什麼都不要印出。

已知最多只有  $n$  列資料，且  $n \leq 100$ ；給定的分子分母都介於  $-1000$  和  $1000$  之間。給定的變數名稱符合 C++ 的變數名稱規範。

給定一筆測試資料，請依序執行每一行指令，並輸出規定的內容。每輸出一次，就換一列；如果該次指令不應該印出任何東西，就直接空一列然後換到下一列。每一列中都不能印出任何上面沒有提到的東西。舉例來說，如果輸入是

```
N x 1 3
N x 10 3
N y2 100 6
P y2
N zf 103 31
I xx
C x zf
```

則輸出應該是

```
50/3

larger
```

## 你上傳的原始碼裡應該包含什麼

你的.cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你**不可以**使用上課沒有教過的方法。

## 評分原則

這一題的其中 40 分會根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。前 30 分由 15 筆測試資料判定分數，一筆測試資料佔 2 分；後 10 分由 5「組」測試資料判定分數，每一組裡面有若干筆測試資料，全對的話才能得到 2 分。

## 第二題

(40 分) 在本題中，我們將實作一個多課程、多學生的成績系統。如果我們有  $m$  個學生、 $n$  個課程，且每個學生在每門課都只會記錄最多一個成績，則最簡單的實作方式是建立一個  $m \times n$  的矩陣，每個矩陣元素是一個成績（或某個值代表沒有修過）。不幸的是，如果  $m$  跟  $n$  都很大，你的程式可能光是處理這個矩陣就會吃不消了。比如說臺大有約 33000 個學生，每學期有約 8000 門課，就算我們只記錄單一學期的成績，也需要  $33000 \times 8000 = 264000000$  個矩陣元素，就算一個元素只用一個 byte 來存，也需要約 264 MB 的記憶體<sup>4</sup>。考慮到每個學生事實上都只會修很少的課，顯然我們應該換個資料結構（也就是存資料的方式）。通常一個學生一學期修不到 10 門課，如果我們只記錄一個學生修的課的成績，應該可以把耗用的空間減少到 0.33 MB 左右。在這一題中，我們就要來試著做這件事。

在這個系統中有兩種東西：學生和課程。我們打算建立一個叫 **Student** 的 class，裡面記錄著一個學生的學生證號、修課數，以及一個將會指向一條動態陣列的指標。這個指標所指向的動態陣列裡，每個元素都是「一門課」，記錄著該門課的課號，以及學生在該門課的成績。為了簡單起見，我們用 C++

<sup>4</sup>如果我們宣告一個 local 的靜態陣列，在很多電腦上這會直接 run-time error。我們當然可以改宣告 global 靜態陣列或 local 動態陣列，但無論如何耗費這麼多空間都不是什麼好主意。

的 `<string>` 裡面的 `string` 來記錄學生證號和課號，用整數來記錄成績（假設我們還活在用百分制記錄成績的年代）。更具體地說，我們可能會宣告一些像這樣的類別（非必要，僅供參考）：

```
class CourseGrade
{
private:
    string courseID;
    int grade;
// something else
};
class Student
{
private:
    string studentID;
    int courseCnt;
    CourseGrade* gradeList;
// something else
};
```

對於學生證號為 `studentID` 的學生，他的 `gradeList` 會指向一個長度為 `courseCnt` 的動態陣列，陣列中的每個元素是一門課的課號 `courseID` 與此學生在此門課的成績 `grade`。這樣一來，我們對每個學生就都只記錄了他修的課的成績了。

你可能有發現這個設計也不是很好。首先，對一個學生要列出他修的所有的課很容易，但對一門課要列出所有修課的學生卻很困難。其次，一門課的課號被大量地儲存在所有修課學生的修課清單中了，但這好像也蠻冗的。在本題中，我們將給你學生、課程與成績的相關資訊，並要求你做一些運算。雖然我們不一定真的會給超大量的測試資料，但請你還是好好想想，是否有更好的設計可以解決上述這些問題？然後請用你自己的設計來寫這個程式吧。

## 輸入輸出格式

系統會提供許多筆測試資料，每筆測試資料裝在一個檔案裡。在每個檔案中，第一列存放一個整數  $n$ ，表示底下共有  $n$  筆資料。在第二列到第  $n + 1$  列中，每一列都存放著一個由兩個逗點隔開的字串，第一個逗點前面是長度為九個字元的學生證號、兩個逗點中間的是長度為七到十字元的課程編號，第二個逗點後面則是介於 0 到 100 之間的成績。課程編號包含一個長度為二到五字元的系所代碼、一個空白鍵，以及一個長度為四個字元的課號。其中系所代碼由大寫英文字母組成，課號則為四位數字。不會有任兩列的學生證號和課程編號組合是重複的。第  $n + 2$  列存放一個字元和一個字串。如果該字元是 `S`，則該字串為一個學生證號；如果該字元是 `C`，則該字串是一個課程編號。給定的學生證號或課程編號一定存在。已知  $1 \leq n \leq 1000$ 。

根據這些輸入，請印出第  $n + 2$  列給定的學生的修課數與平均分數（無條件捨去到整數位），或者給定的課程的修課人數與平均分數（無條件捨去到整數位）。兩個印出的數字中間用一個空白鍵隔開。舉例來說，如果輸入是

```
10
B90705023,IM 1001,90
```

```
B90705023,IM 1002,80
B90705023,IM 1003,82
B90705023,CS 2011,86
B90705023,ECON 2011,90
B90705023,ECON 2012,69
R04725008,ACCT 1001,76
R04725008,IM 1002,75
R04725008,IM 2010,82
R04725008,PHYED 1001,88
S B90705023
```

則輸出應該是

```
6 82
```

如果輸入的最後一列變成

```
C IM 1002
```

則輸出應該是

```
2 77
```

## 你上傳的原始碼裡應該包含什麼

你的.cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用任何方法。

## 評分原則

- 這一題的其中 40 分會根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。前 30 分由 15 筆測試資料判定分數，一筆測試資料佔 2 分；後 5 分由 5「組」測試資料判定分數，每一組裡面有若干筆測試資料，全對的話才能得到 2 分。
- 這一題的其中 20 分會在作業十中被評定。屆時我們會讓同學們互相檢視彼此的本題程式碼，並且就可讀性、易維護性、模組化程度、排版等面向寫評語和給評分（當然一切都是匿名的）。該任務在本題中會佔 20 分，其中 10 分取決於檢視你的程式碼的同學給你的分數（必要時助教會出來主持公道，請不用緊張），另外 10 分取決於你對同學的程式碼的評語和評分的合理性和建設性。若你在本次作業中沒有寫這一題，那屆時自然沒有人能檢視你的程式碼，你也就得要損失這 10 分了。

## 第三題

(0 分) 請在 PDOGS 上批改你被隨機分配到的作業八的程式碼，根據它在正確性以外的部份給它 1 至 5 分的評分，並且說明你給分的依據。建議在評分時參考以下五個面向，一個面向上做得好就得一分，

還不錯則半分，不好則零分，五個面向的分數合計後無條件進入即為你最後給的總分。

- 可讀性：變數與函數名稱是否具有合適的資訊量？程式碼排版是否良好且具有前後一致性？是否有合適的註解？關於註解，當然不需要每一行都有註解，但若你發現在某一大段落裡都沒有註解，或某個你感覺很不易看懂的部份沒有註解，你可以指出來；不要直接說「註解太少」但沒有說是哪邊缺乏註解。
- 模組化程度：是否有宣告合適的函數、structure 或 class？是否有避免將非常類似的程式片段寫複數次而非寫成函數？是否有避免一個函數做非常多事情？直接閱讀 main function 是否能很快地理解程式在大方向上的運算邏輯？
- 效率：程式運算是否有合理的運算效率？當然我們不要求每個同學都寫出超級有效率的精妙演算法，但至少一個程式不應該進行過多不必要的運算，也不應該耗用過多不必要的記憶體空間。如果你看不出這個程式的效率有明顯的問題，我們建議你直接給一分。
- 擴充性：當要解的問題變得更複雜的時候，我們能不能簡單地修改這個程式以解決新的問題，而不是寧可砍掉重練？這個議題當然也很主觀，所以如果你不能明確地指出在怎樣的新問題上，這個程式會有擴充性問題，我們建議你直接給一分；如果你不能指出很嚴重的問題，我們建議你至少給半分。但對批改者來說，這個關於擴充性的思考其實是很好的訓練。試試看吧！
- 其他：如果有任何其他令你想扣分的理由，請明確地寫出來並且在這個面向上扣分；沒有的話就給一分。

## 第四題 (bonus)

(20 分，各 5 分) 請根據第一題和第二題的題目回答下列問題：

- (a) 我們是否應該為 `RationalNumber` 實作 copy constructor 或 destructor，或者 overload assignment operator？為什麼？
- (b) `Student` 滿足一些讓我們為其實作 copy constructor、destructor 和 assignment operator 的理由，但實際應用上我們既不會複製一個學生，也不會把一個學生 assign 成另一個學生。如果我想要避免 default assignment operator 被呼叫，但又不想實作一個 assignment operator（因為根本不應該提供這個功能），我應該怎麼做？請把程式碼寫下來並且說明原因。
- (c) 承上題，請實作 `Student` 的 destructor。
- (d) 承上題，事實上我們是需要自己實作 copy constructor 的，因為「應該用得到」。請說明在學校裡事實上沒有複製人的情況下，`Student` 的 copy constructor 何時會被用到？接著請實作這個 copy constructor 去實現 deep copy。