

程式設計 (113-1)

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/>) 為第一、二、三題上傳一份 C++ 原始碼 (以複製貼上原始碼的方式上傳)，並且到 NTU COOL 上傳一份 PDF 檔。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。這份作業的程式碼和書面報告的截止時間都是 **11 月 19 日早上八點**。為這份作業設計測試資料並且提供解答的助教是彭麒任。

在你開始前，請閱讀課本的第 8 和 18 章 (分別關於 File I/O 以及 C++ string)¹。

本次作業滿分為 110 分，得幾分就算幾分。若整學期有 n 份作業，則學期的作業總成績即為 n 份作業的總分除以 n (不論超過 100 與否)。

第一題

(20 分) 本題延續自作業八的第二題，大部份情境和設定與該題一樣，但有一些細節與之前的不盡相同，也有新增新的要求。以下將從頭完整地敘述本題，建議大家在此都還是要詳細閱讀一次題目。

有一家診所，該診所所有兩位醫師，編號分別為 1 和 2，且兩位醫師的看診速度相同。編號 2 的醫師要兼顧管理工作，所以在診所剛開門的時候不看診，而是在第 V 分鐘時才加入看診。在醫師 2 投入看診前，所有患者都會去醫師 1 的診間接受看診或排隊。每位患者都有各自的年齡、抵達時間和看診時間長。我們稱呼第 i 個抵達的患者為患者 i ，而患者 i 的年齡為 g_i 、抵達時間為 a_i 、看診時間長為 s_i 。這間診所不想讓年長者等太久，所以當一位患者抵達時，診所會引導他詢問排在他前面一位的患者的年齡，如果他比他前面一位年長至少 b 歲 (包含 b 歲) 時，診所會讓這位年長患者跟前面這位相對年輕的患者交換序位，然後再繼續往前看是否可以繼續交換，直到不能交換為止²。無論如何，都只能跟等待中的患者交換位置；已經在接受看診的患者無論如何都會被看完，不會被中斷。

在第 V 分鐘，醫師 2 投入看診的瞬間，有以下各種可能：

- 如果沒有患者在醫師 1 的診間排隊，則在患者方面沒有任何變化。
- 如果有若干位患者正在醫師 1 的診間排隊，則原本在醫師 1 的診間中排第一位的患者會立刻給醫師 2 看診、原本在醫師 1 的診間中排第三位的患者會到醫師 2 的診間排第一位、原本在醫師 1 的診間中排第五位的患者會到醫師 2 的診間排第三位，依此類推；原本在醫師 1 的診間中排第二、四、六位的患者會繼續停留在醫師 1 的診間，變成排第一、二、三位，依此類推。這種情況下的患者排隊狀態變化如圖 1 所示。
- 如果在那個瞬間，剛好醫師 1 那邊有一位患者被看診結束並且離開診所，則在醫師 1 的診間中原本排第一位的患者會立刻給醫師 1 看診、排第二位的患者會立刻給醫師 2 看診、排第三位的患者會在醫師 1 的診間排第一位、排第四位的患者會在醫師 2 的診間排第一位、排第四位的患者會在醫師 1 的診間排第二位、排第五位的患者會在醫師 2 的診間排第二位，依此類推。這種情況下的患者排隊狀態變化如圖 2 所示。

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

²這個流程很像 insertion sort 中找到正確位置插入的流程。

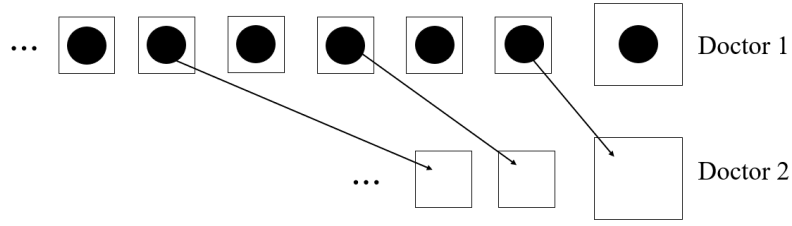


Figure 1: 醫師 2 加入時醫師 1 持續看診的排隊狀態變化

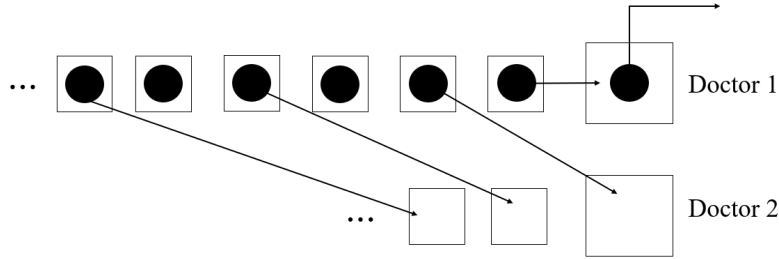


Figure 2: 醫師 2 加入時醫師 1 恰好看診完成的排隊狀態變化

從第 V 分鐘起（不包含第 V 分鐘），如果有一位新患者出現，診所會檢視兩個診間的狀況，然後將他安排到總人數（包含被看診中和排隊中）較少的診間；若兩個診間的總人數相同，則將該患者安排至醫師 1 的診間中。這位患者接著就按照第 V 分鐘前的規則在該診間排隊、交換排隊順序、接受看診。

在第 V 分鐘那個瞬間有許多事件可能同時發生，包括醫師 1 完成一位患者的看診、醫師 2 加入，以及新患者抵達。如果在第 V 分鐘那個瞬間有以上三個事件的任意兩個以上發生，我們說這數個事件都發生在第 V 分鐘那個瞬間，但還是有先後順序，依序是新患者抵達並且進入醫師 1 的診間排隊或就診並且完成必要的排隊順序交換，接著是醫師 1 完成一位患者的看診，最後是醫師 2 加入。

根據以上資訊，我們可以推算每位病患在每個時間點的狀態。具體來說，對患者 i 我們可以推算他被哪位醫師看診、在該診間的看診序號、開始接受看診時間、看診完成時間 f_i 、等待時間長 w_i ，以及患者 i 抵達的前一瞬間的整個診所（包含兩個診間）內人數 p_i （因為是前一瞬間，所以不包含他自己；診所中的人數包含被看診中的和等待中的）。

我們用以下的例子展示前述規則。假設患者們的資訊如表 1 所示， $b = 10$ ，醫師 2 在第 $V = 12$ 分鐘加入：

- 患者 1 到達時由醫師 1 看診，看診完成時間為第 $f_1 = 6 + 5 = 11$ 分鐘。
- 患者 2、3、4、5 依序在第 7、8、9、10 分鐘到達。他們到達時，醫師編號 1 在為患者 1 看診，因此他們四位都需要排隊。患者 2 比患者 1 年長許多，但抵達時患者 1 已經在接受醫師 1 看診了，因此患者 2 無法跟患者 1 交換。而 $g_4 \geq g_3 + 10$ ，意即患者 4 比比他早到的患者 3 年長 10 歲以上，因此患者 4 會跟患者 3 交換看診序號；又因為 $g_5 \geq g_3 + 10$ ，患者 5 抵達時會先跟患者 3 交換看診序號，再因為 $g_5 \geq g_4 + 10$ ，患者 5 接著再跟患者 4 交換排隊順序。最終結果是在第 10 分鐘時，患者 2、3、4、5 在醫師 1 的診間會依次排在第一、四、三、二位。

項目	抵達順序 i					
	1	2	3	4	5	6
年齡 g_i	38	66	34	52	65	80
到達時間 a_i	6	7	8	9	10	25
看診時間長 s_i	5	8	4	6	2	3

Table 1: 患者資訊

- 患者 1 在第 11 分鐘離開後，醫師 1 繼續為患者 2 看診，患者 2 的看診完成時間為第 $f_2 = 11 + 8 = 19$ 分鐘，等待時間為 $w_2 = 11 - 7 = 4$ 。
- 在第 $V = 12$ 分鐘，醫師 2 加入。當時正在醫師 1 的診間排第一位的是患者 5、排第二位的是患者 4、排第三位的是患者 3，因此患者 5 會在第 12 分鐘由醫師 2 看診，患者 5 的看診完成時間為第 $f_5 = 12 + 2 = 14$ 分鐘，等待時間為 $w_5 = 12 - 10 = 2$ ，而患者 4 會在醫師 1 的診間中排第一位、患者 3 會在醫師 2 的診間中排第一位。
- 患者 2 在第 19 分鐘離開後，醫師 1 會繼續為患者 4 看診，患者 4 的等待時間 $w_4 = 19 - 9 = 10$ ，看診完成時間則為第 $f_4 = 19 + 6 = 25$ 分鐘。
- 患者 5 在第 14 分鐘離開後，醫師 2 會繼續為患者 3 看診，患者 3 的看診完成時間為第 $f_3 = 14 + 4 = 18$ 分鐘，等待時間為 $w_3 = 14 - 8 = 6$ 分鐘。
- 患者 6 於第 25 分鐘到達時，兩位醫師都在閒置，因此診所會安排由醫師編號 1 為他看診，患者 6 的看診完成時間則是 $f_6 = 25 + 3 = 28$ 。

上述推算資訊可以被整理如表 2。

項目	抵達順序 i					
	1	2	3	4	5	6
為其看診之醫師編號	1	1	2	1	2	1
在醫師 1 的看診序號	1	2	–	3	–	4
在醫師 2 的看診序號	–	–	2	–	1	–
開始接受看診時間	6	11	14	19	12	25
看診完成時間 f_i	11	19	18	25	14	28
等待時間長 w_i	0	4	6	10	2	0
患者 i 到達的診所內人數 p_i	0	1	2	3	4	0

Table 2: 推算結果

在本題中，診所希望在任意指定一個抵達順序 k 之後，你可以幫他們找出患者 k 的看診完成時間 f_k 、從抵達診所到接受看診的等待時間長 w_k ，以及他到達診所的前一瞬間診所中的總人數 p_k 。診所也希望在給定時間點 t 後，你可以幫他們算出在第 t 分鐘那個瞬間的前一瞬間的診所中的總人數 $x(t)$ 。

輸入輸出格式

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有五列，其中第一列裝著三個正整數，依序是 n 、 b 和 V ；在第二列裝著 n 個正整數，依序是 g_1 、 g_2 、 g_3 直到 g_n ；在第三列裝著 n 個正整數，依序是 a_1 、 a_2 、 a_3 直到 a_n ；第四列裝著 n 個正整數，依序是 s_1 、 s_2 、 s_3 直到 s_n ；第五列裝著一個正整數 k 、一個逗點、一個正整數 t 。已知 $1 \leq n \leq 100$ 、 $0 \leq b \leq 50$ 、 $0 \leq V \leq 10000$ 、 $10 \leq g_i \leq 100$ 、 $1 \leq a_1 < a_2 < a_3 < \dots < a_n \leq 100000$ 、 $1 \leq s_i \leq 1000$ 、 $k \in \{1, 2, \dots, n\}$ 、 $1 \leq t \leq 10000$ 。

讀入這些資訊後，請依照題目指定的規則計算 f_k 、 w_k 、 p_k 和 $x(t)$ ，並將這些數字依序印出，相鄰的兩個數字中間用一個逗點隔開。舉例來說，如果輸入是

```
6 30 20
38 37 34 52 47 25
6 15 18 19 27 38
5 8 4 6 2 3
3,20
```

則輸出應該是

```
24,2,1,3
```

如果輸入是

```
6 10 10
38 37 34 52 65 80
6 7 8 9 19 25
5 8 4 6 2 3
4,16
```

則輸出應該是

```
16,1,3,3
```

如果輸入是

```
6 5 18
20 25 30 35 40 45
10 11 12 13 14 15
2 4 5 6 5 5
5,25
```

則輸出應該是

```
23,4,3,2
```

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只包含指定的函數 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

這一題的所有分數都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。一筆測試資料佔 2 分。

第二題

(20 分) 有一個國家，該國國民的身分證字號共 8 碼，在本題中我們稱這第 i 碼為 x_i 。第一碼 x_1 是不包含 I 和 O 的大寫英文字母，接著六碼 x_2, x_3 直到 x_7 各是不包含 4 在內的數字。最後一碼 x_8 則是檢查碼，如果 x_1 是 A 到 H，則 $x_2 + \dots + x_8$ 除以 10 的餘數應該是 0；如果 x_1 是 J 到 Q，則 $x_2 + \dots + x_8$ 除以 10 的餘數應該是 5；如果 x_1 是 R 到 Z，則 $x_2 + \dots + x_8$ 除以 10 的餘數應該是 9。

在本題中，給定數個字串，請判斷它們是不是合法的身分證字號。舉例來說，「A87123」不是，因為長度不對；「A87 0123」不是，因為含有非英文和數字字元；「12395678」不是，因為第一個字元不是英文；「A1112222」不是，因為後面七碼加起來除以 10 的餘數不是 0；「O1112221」不是，因為第一碼是 O；「B1141111」不是，因為含有 4；「j1112226」不是，因為 j 是小寫的；「J1112226」則合法。

輸入輸出格式

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在檔案中會有 $n + 1$ 列，前 n 列中的每一列有一個包含英文字母、數字、空白字元的字串，最後一列則是一個單獨的英文句號。已知一個字串長度最長為 100 個字元，且 $1 \leq n \leq 10$ 。

讀入這些資訊後，請判斷前 n 列的每一列是否是一個合法身分證字號，並且輸出 n 個「Y」或「N」，若第 j 列是合法的就輸出「Y」，不然就輸出「N」。舉例來說，如果輸入是

```
A87123
A87 0123
12395678
A1112222
O1112221
B1141111
j1112226
J1112226
.
```

則輸出應該是

```
NNNNNNNY
```

如果輸入是

```
J1112226
.
```

則輸出應該是

```
Y
```

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

這一題的所有分數都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。一筆測試資料佔 2 分。

第三題

(50 分) 本題為作業七第一題的延伸題目。爲了讓大家不用回頭看題目，底下我們重新敘述這個題目一次。

奧運會上各國之間會有獎牌排名，雖然有各種排名方式，但最常見的是「先比金牌數，多者排前；如果金牌數一樣，則比銀牌數，多者排前；如果銀牌數一樣，則比銅牌數，多者排前；如果三個獎牌數都一樣，則會比哪個國家先拿到該國獲得的最後一面金牌，先者排前」。

在一次賽會中，大會按照頒發獎牌的先後列出了各個比賽項目得獎牌的國家，每一列代表一個比賽項目，排在每一列的最前面的是得金牌的國家，排在中間的是得銀牌的國家，排在最後面的是該項目得銅牌的國家，中間以逗點隔開。一個國家也可以在一項比賽之中同時贏得多個獎牌。舉例來說，某次賽會的獎牌清單如下（請參見 hw09_countriesList.txt）：

```
United States,Canada,Brazil
Germany,France,Italy
Japan,    South Korea,Australia
United Kingdom, China,India
United States,South Africa, Mexico
Canada,United States,Germany
Brazil,Japan,    France
Italy,United Kingdom,Southkorea
Australia,China, United States
Canada, Mexico, SouthAfrica
BrAZIL,Brazil,    Japan
Russia,Italy,Canada
```

South Korea, Germany, AUstralia UnitedStates, France, IndiA BrAZil, Canada, Russia
--

表示第一個頒獎的比賽中獲得金、銀、銅牌的國家依序是美國、加拿大、巴西，第二個頒獎的比賽中獲得金、銀、銅牌的國家依序是德國、法國、義大利，依此類推。

如大家所見，大會提供的清單有些問題。請使用以下規則處理：

- 國家名稱的前後的空格是多餘的，應該一律被清除，例如前面範例中第三次頒獎得銀牌的「 South Korea 」的前面的空格應該被清除，該國家的國名應該被處理成「South Korea」，再繼續後續處理流程。
- 同一個國家的名稱中間可能有為數不同的空格，每一組連續的空格都應該被刪到只剩下一個，例如「United States」應該被處理成「United States」。如果有國家的名稱中有複數組連續空格，請將每組連續空格都刪到只剩下一個，例如「Republic of Ireland」應該被處理成「Republic of Ireland」。
- 國家名稱是否有空格是有差別的，其他字元都相同但一個中間有空格與一個中間沒有空格的國家名稱，應該被視為兩個國家，例如「UniteD States」和「UnitedStates」應該被視為不同。
- 同一個國家的名稱可能有大小寫的不同，但應該被視為同一個國家，例如「InDia」和「IndiA」應該被視為相同。如果最後要輸出某個國家的國名，其大小寫應以該國第一次出現在清單中的名稱為代表。以印度為例，在此範例中請輸出「InDia」。

請按照上述規則處理這份清單，最終幫所有國家計算它們的得牌情況，並且輸出前三名的國家名稱以及它們的金牌、銀牌、銅牌數量。

特別說明 1：在本題中，我們要求大家建立一個 `Country` 類別來記錄每個國家的名稱及其金、銀、銅牌數量，在讀取每一次頒獎的結果後，根據獲得金、銀、銅牌的國家名稱來更新對應的 `Country` 物件的獎牌數。為了練習，我們要請大家在 `Country` 類別中重載 `operator<` 運算子，去進行各國之間的排名。

特別說明 2：在本題中，大家無法預先確定檔案中的國家數量和頒獎次數，因此我們建議大家建立一個動態指標陣列，每個指標指向一個國家物件，每當多一個新的國家獲得獎牌，就新建一個物件並且讓陣列中的一個指標指向該國家。如果你的陣列長度不夠了，就另外動態宣告一個更長的陣列（例如原本陣列的兩倍長），讓新陣列的部份指標指向那些國家物件，然後清除舊的陣列。這樣一來，你的程式就可以動態地處理不確定數量的國家。

特別說明 3：這個題目純粹是爲了讓同學們練習字串處理，且剛好情境中的字串都是國家名稱，所以會出現一些國家名稱的大小寫、空格長得亂七八糟的情況，實屬不得已。出題者並沒有任何對這些國家不敬的意思。

輸入輸出格式

系統會提供一共 15 組測試資料，每組測試資料裝在一個檔案裡。在檔案中會有一列是每次頒獎獲得獎牌的國家清單的檔案路徑。清單內會有 n 列，每列有三個國家名稱，兩兩之間用逗點隔開，國家名稱只包含大小寫英文字母和空白字元。已知一個國家名稱包含 x 個字元，且 $1 \leq x \leq 100$ 。另外已知 $1 \leq n \leq 1000000$ ，且至少有三個不同的國家。

讀入檔案字串後，請把這個字串不做任何修改、直接傳進你用來讀取檔案的 constructor 或函數中，就可以讀取已經被放在 PDOGS 上的檔案。接著請讀取名單內容，並輸出前三名國家與其各獎牌數量。舉例來說，如果輸入是

```
/assisting_data/hw09_countriesList.txt
```

且這個檔案的內容如附件（請假設這個檔案在 PDOGS 上放置的路徑是在 assisting_data 底下），則輸出應該是

```
BRazil:3,1,1
Canada:2,2,1
United States:2,1,1
```

請注意在這個例子中，有個 `UnitedStates` 得了一面金牌，但這面金牌不應該被計入 `United States` 的獎牌裡。

在另一個例子裡，如果輸入是

```
/assisting_data/hw09_countriesList2.txt
```

則輸出應該是

```
United States:3,1,1
BRazil:3,1,1
Canada:2,2,1
```

請注意在這個例子中，倒數第二次頒獎得到金牌的「 `United States` 」應該被視為 `United States`，且 `United States` 比 `BRazil` 更早得到它得到的最後一面金牌，所以 `United States` 排名在前。

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：

- 確定可以使用的語法包含 C++ 字串、自訂標頭檔、`<fstream>` 裡面的所有功能，以及在之前的作業中正面表列過的語法。
- 確定不可以使用的語法包含 `printf`、`scanf`、繼承、多型等等。

請注意正面表列的固然是都確定可以用，但沒有被負面表列的不表示可以用喔！

評分原則

- 這一題的其中 30 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。一筆測試資料佔 2 分。

- 這一題的其中 20 分會根據你所寫的程式的品質來給分。助教會打開你的程式碼並檢閱你的程式的結構、運算邏輯、可讀性（包含排版、變數命名、註解等等）、可擴充性、模組化程度，以及是否使用了還沒教過的語法。請寫一個「好」的程式吧！

第四題

特別說明：本題為作文題而非程式題，請不要上傳東西到 PDOGS；請把你的答案放在 PDF 檔中上傳到 COOL。由於班上同學很多，但助教人力有限，所以原則上我們會從所有繳交中隨機批改一部份（可能是 50%、90%、100%，但不會低於 50%）。沒被抽到的同學如果有認真回答，在本題就會得到滿分，但不會得到直接的助教回饋，如果沒回答或隨便亂寫當然就不會得分；有被抽到的同學們可能會被扣分，但相對應地也會得到助教的回饋。

（20 分）在第三題中，你應該寫了不少類別、函數、常數等等。請寫一個或數個自己的 .h 檔，把你認為合適的程式碼放進去，再把剩下的程式碼放在一個或數個 .cpp 檔。請確保你分割完程式碼之後，這些程式碼在你的開發環境中是可以被正確編譯和執行的。請畫一張圖說明你的程式架構，用一個框框代表一個檔案或一個 C++ 的標準函式庫（例如 `iostream` 那些），並且用 $A \rightarrow B$ 表示 B 有 `include A`。在每一個代表你寫的檔案的框框裡面或旁邊寫下該檔案裡面宣告的類別、函數、常數（只要寫在「宣告」該類別、函數、常數的地方就好，不用寫在「定義」該類別、函數、常數的地方）。最後，請在你的架構圖後面用成段落的文字說明你設計這樣的程式架構（將這些類別、函數、常數這樣分割到不同檔案）的設計理念。