商管程式設計（107-2）
作業八

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/`）為第一題做同儕互評，再為第二題至第四題各上傳一份 Python 3 原始碼（以複製貼上原始碼的方式上傳）。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 ***5 月 20 日晚上九點***。在你開始前，請閱讀課本的第十四章第一至四節以及第十五至十七章[1]。為這份作業設計測試資料並且提供解答的助教是陳潔智。

第一題

（20 分）請在 PDOGS 上批改你被隨機分配到的作業七第三題的程式碼，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六個面向。在前五個面向上，一個面向上做得好就得一分，還不錯則半分，不好則零分；在第六個面向上則在有必要時扣分。六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？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？函數間是否有合適的 decoupling？直接閱讀程式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。
- 題目規範：你應該檢查那份程式碼有沒有違反題目的規範，如果有（例如題目說不可以用上課沒教過的東西，但他用了，或者題目說一定要寫個函數，但他沒寫），就扣他三分。當然，請明確地指出他哪邊違反了題目的規範。

[1] 課本是 A. Downey 所著的 *Think Python 2*，在 `http://greenteapress.com/wp/think-python-2e/` 可以下載。

本題其中 10 分取決於檢視你的程式碼的同學給你的分數總和（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（原則上除非被申訴，且助教檢視後發現你確實評得很不公允，否則只要有評就會得到 10 分）。

特別附註：在作業七時，其實只有留 12 分是互評的分數，但這裡一共有 20 分後，所以作業七相當於是滿分 108 分了。

第二題

（20 分）在 Python 中，我們可以使用 `random` 函式庫中的 `random()` 函數，來產生介於 0 到 1 之間的隨機亂數。舉例來說：

```
import random
print(random.random())
```

就會印出一個介於 0 到 1 之間的隨機小數。當然，每次你執行這個程式的時候，結果都會不太一樣。假設你想要印出介於 0 到 100 之間的隨機整數，大體上來說只要自行處理 `random()` 回傳的值即可，例如這樣：

```
import random
print(int(random.random() * 100))
```

但電腦是怎麼產出亂數的呢？事實上大部份的電腦系統中，都是用演算法來產出「假亂數」（pseudo-random number）。以下我們用產生整數來做說明。最常見的作法是，系統中先設定好一個亂數種子（random number seed）x_0，以及三個正整數 a、b 和 c。接著第一個亂數 $x_1 = \mathrm{mod}(ax_0 + b, c)$，（$\mathrm{mod}(y, z)$ 是把 y 除以 z 所得的餘數），第二個亂數 $x_2 = \mathrm{mod}(ax_1 + b, c)$，依此類推。舉例來說，如果 $x_0 = 1$、$a = 7$、$b = 11$、$c = 13$，則依序我們會得到 $x_1 = 5$、$x_2 = 7$、$x_3 = 8$、$x_4 = 2$，依此類推。很顯然地，這樣做只能產生介於 0 到 12 之間的亂數，所以實務上 c 會是很大的整數，a 跟 b 自然也就得要很大了。此外，如果 a、b、c 的值設得不好，這串亂數就會很「不亂」，例如如果 $x_0 = 1$、$a = 7$、$b = 11$、$c = 17$，我們就會得到 $x_1 = x_2 = x_3 = \cdots = 1$ 了。

在本題中，我們將請你實做一個更簡單、更無理 (?) 的亂數產生器。我們的每個亂數 x_k，會是一個最多八位數的非負整數。給定 x_k 後，我們將之平方，再取其第十二位數到第五位數，當作 x_{k+1}。例如若 $x_k = 12345678$，那 $x_k^2 = 152415765279684$ ，第十二位數到第五位數為 41576527，這就是 x_{k+1}。如果要輸出的數字的開頭若干位數剛好是 0，則不要輸出那些 0，例如針對 00123405 請輸出 123405。

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。在每個檔案中會有一行，包含兩個整數 x_0 與 n。兩個輸入的數字之間用一個逗點隔開。已知 $1 \le n \le 20$、$0 \le x_0 \le 99999999$。讀入這兩個數字後，請按照題目指定的規則，依序輸出 x_1、x_2 直到 x_n。兩個輸出的數字之間用一個逗點隔開。

舉例來說，如果輸入是

```
12345678,3
```

則輸出應該是

```
41576527,60759738,74576182
```

如果輸入是

```
10000000,4
```

則輸出應該是

```
0,0,0,0
```

針對這個題目，你***可以***使用任何方法。這一題的 20 分都根據程式運算的正確性給分，一筆測試資料佔 2 分。

附註：這一題當然是很簡單，幾乎是送分，但如果你有興趣，可以自行想想看，這方法產出的亂數是否真的很亂？

第三題

（60 分）在本題中，我們想寫一個簡單的一日行事曆（schedule）。我們將會讀入許多行程（event），每個行程都有四個資訊：行程名稱、開始時間（精確到秒）、結束時間（精確到秒）、價值。每當我們讀入一個行程，我們就嘗試將此行程排入當日的行程表中。如果沒有時間衝突，就直接排入；如果正在被考慮的 A 行程跟已經在行程表上的 B、C、D 等行程衝突，就比較 A 的價值和所有跟它衝突的行程的價值總和，前者較大就刪除既有的衝突行程並且排入 A，反之則放棄 A。我們假設行程間沒有移動時間，只要前一個行程的結束時間不晚於後一個行程的開始時間，就視為不衝突。例如 12:10:00 結束的 A 行程跟 12:10:00 開始的 B 行程視為不衝突。

安排完整個行程後，請印出所有被排入行程表的行程個數，以及它們的總價值。

輸入輸出格式

系統會提供一共 20 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 $n+1$ 行，第一行包含一個整數 n，第二行起包含 n 個行程的資訊，每一行是一個行程，依序是行程的名稱、開始時間、結束時間和價值，任兩個資訊之間用一個逗點隔開。已知名稱是大小寫英文字母、數字與空白字元組成的字串，長度最多 50 字元，且名稱的頭尾字元一定不是空白；兩個時間都是以 hh:mm:ss 格式呈現，結束時間必然晚於開始時間；價值是 1 到 100 之間的整數；逗號左右一定不會是空白。

讀入資料後，請按照題目的規定安排行程，最後印出被排入的行程個數與其總價值，兩個整數之間用一個逗點隔開。舉例來說，如果輸入是

```
3
Programming FOR Business Computing,09:10:00,12:10:00,30
Introduction to Computer Science,13:20:00,15:20:00,8
Dating with my lover,18:30:30,20:00:00,100
```

則輸出應該是

```
3,138
```

如果輸入是

```
4
Sleep,09:00:00,16:00:00,50
Programming FOR Business Computing,09:10:00,12:10:00,30
Introduction to Computer Science,13:20:00,15:20:00,38
Dating with my lover,18:30:30,20:00:00,100
```

則輸出應該是

```
2 150
```

應注意的是，雖然中間兩個行程的總價值（$30 + 38 = 68$）超過第一個行程的價值，但根據演算法，我們會排第一個行程，接著第二個行程無法取代第一個，第三個也無法取代第一個，只有第四個順利被排入了。

你上傳的原始碼裡應該包含什麼

你的.py 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 Python 3 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，但上課介紹過的函式庫中所有的功能都可以用。

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 會直譯並執行你的程式、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 分會在作業九中被評定。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。若你在本次作業中完全沒有寫這一題，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。

  此外，由於本作業是練習使用 class，所以我們在此要求你必須要寫 class 並且實做題目中要求的那兩個函數，然後適當地使用它們。如果屆時你負責評分的程式沒有做到這件事，請給他 0 分；如果你的程式沒做到這件事，你也就會損失本題這 10 分了。

第四題（bonus）

（20 分）承上題，在讀完行程並且安排好行程表後，現在我們想要搜尋這個行程表上的行程。本題的輸入會跟第三題幾乎一樣，但最後面多一行，包含一個長度最多為 50 個字串 s。程式應該找出所有被排入的行程中有完整地包含 s 當子字串的行程，並依照開始時間順序由早到晚依序逐行印出，印出的格式和讀入時一樣。如果沒有任何行程被搜尋到，就印出一行「Nothing found.」，前後無空格、無換行。判定是否為子字串時，大小寫字母視為相同。

舉例來說，如果輸入是

```
4
Introduction to Computer Science,13:20:00,15:20:00,8
Programming FOR Business Computing,09:10:00,12:10:00,30
Dating with my lover,18:30:30,20:00:00,100
Fixing Computer,00:00:00,23:00:00,1
comput
```

則輸出應該是

```
Programming FOR Business Computing,09:10:00,12:10:00,30
Introduction to Computer Science,13:20:00,15:20:00,8
```

請注意 Fixing Computer 不應該被印出，即使它含有「comput」這個子字串，因為 Fixing Computer 沒有被排入，因此就不應該被搜尋到。如果輸入是

```
4
Introduction to Computer Science,13:20:00,15:20:00,8
Programming FOR Business Computing,09:10:00,12:10:00,30
Dating with my lover,18:30:30,20:00:00,100
Fixing Computer,00:00:00,23:00:00,1
hahaha
```

則輸出應該是

```
Nothing found.
```

針對這個題目，你***可以***使用任何方法。這一題的 20 分都根據程式運算的正確性給分，一筆測試資料佔 2 分。