

商管程式設計 (106-1)

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/>) 為第一題上傳一個 PDF 檔、為第二題做同儕互評，再為第三題與第四題各上傳一份 Python 3.6 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 **2017 年 12 月 4 日凌晨一點**。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鄭允頌。

第一題

(20 分) 針對以下題目，請寫 python 程式去畫出題目要求的圖。你可以不使用 `matplotlib`，但你一定要寫 python。請貼上你的 python 程式和畫出來的圖到 PDF 檔中。

- (a) (10 分) 在本週課程使用的「ubike.csv」檔中，我們有每個車站的 latitude 和 longitude 資訊。請讀取這個檔案，取出所有車站的 latitude 和 longitude 資訊，以前者為 y 軸座標、後者為 x 軸座標，將這些車站的位置標示在一個散佈圖上。請把圖形的 x 座標範圍設定在 121.52 到 121.57 之間、 y 座標設定在 25.01 到 25.05 之間。
- (b) 承上題，請在畫出這些車站的位置時，考慮各站點間的距離。首先，在計算距離時，請用課程投影片中介紹的函數 `haversine`：

```
import math

def haversine(lat1, lon1, lat2, lon2):
    lon1, lat1, lon2, lat2 = map(math.radians, [lon1, lat1, lon2, lat2])
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = math.sin(dlat / 2) ** 2
    a += math.cos(lat1) * math.cos(lat2) * math.sin(dlon / 2) ** 2
    return 6367 * (2 * math.asin(math.sqrt(a)))
```

請用這個函數，找出所有車站兩兩之間的距離，進而找出哪兩個車站間的距離是所有距離中最長的。找到那兩個車站之後，請在散佈圖上把那兩個車站的點畫成紅色，其他車站的點則畫成藍色。

第二題

(20 分) 請在 PDOGS 上批改你被隨機分配到的作業七第一題的程式碼，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

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

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，或者題目說一定要用指標和動態記憶體配置，但他沒用），就扣他三分。當然，請明確地指出他哪邊違反了題目的規範。

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）

第三題

（60 分）在本週課程使用的「midterm2.csv」檔中，我們記錄了本門課某學期第二次期中考的所有提交記錄。檔案共有七個欄位。「SubmissionID」是每次提交程式碼時系統給定的唯一編號、「StudentID」是那次提交程式碼的學生編號（一個學號對應到一個學生編號，但你從學生編號看不出學號）、「Problem」是提交的題號（1 到 4 的整數）、「Status」是提交後的狀態（Accepted、Compile Error、Runtime Error、Time Limit Exceed、Wrong Answer 五種）、「Score」是該題的得分（第一題到第四題的滿分各是 30、40、30、30，考試總分 130 分）、「CodeLength」是該次提交的程式碼的長度、「SubmissionTime」是提交的時間。請注意這個檔案中的資料列是依提交時間由晚到早排序的。

在本題中，我們會把這個檔案放在 PDOGS 上，再告訴你這個檔案的絕對路徑，要求你上傳的程式碼去讀取這個檔案。我們將給你一個開始時間和結束時間，請你計算這段時間中（包含這兩個瞬間）每一題的五種狀態各發生幾次。舉例來說，如果給定 10:00:00 到 10:20:00，則我們共有 33 筆提交，資訊整理如表 1。請印出這 20 個數字。

題目	Accepted	Compile Error	Runtime Error	Time Limit Exceed	Wrong Answer
1	7	0	0	0	3
2	1	0	0	0	2
3	4	0	3	0	4
4	4	0	4	0	1

表 1: 範例一

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。在每個檔案中會有兩行，第一行包含一個字串，代表需讀取之檔案的絕對路徑；第二行包含兩個字串，依序代表指定的開始時間與結束時間（所以前者早於後者）時間的格式為 `hh:mm:ss`，例如九點十八分零六秒表示為 `09:18:06`。兩個時間字串之間被一個空白隔開。我們只會給你隨本題附上的這個「`midterm2.csv`」檔案。

讀入資料後，請按照題目的規定計算出每一題在兩個時間（含）之間的五種狀態的提交次數，接著以四列印出，第一列印出第一題、第二列印出第二題，依此類推。在每一列中，請依序印出該題 Accepted、Compile Error、Runtime Error、Time Limit Exceed、Wrong Answer 的次數。每兩個數字之間用一個空白鍵隔開。每一列最後都應該有一個換行字元（包含最後一列），但每一列的最後一個數字後面不可以有空格。舉例來說，如果輸入是

```
C:/thisIsAnExamplePath/midterm2.csv
10:00:00 10:20:00
```

則輸出應該是

```
7 0 0 0 3
1 0 0 0 2
4 0 3 0 4
4 0 4 0 1
```

如果輸入是

```
C:/thisIsAnExamplePath/midterm2.csv
10:19:45 10:19:45
```

則輸出應該是

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
```

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 3.6 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。

第四題（bonus）

（20 分）承上題，我們繼續對「midterm2.csv」做運算。給定一個學生編號後，我們可以找出這個學生的全部提交記錄，以及每一題的最後一次提交的時間。舉例來說，編號 19 的學生在第一題是 10:57:05、第二題是 11:38:10、第三題是 12:06:24、第四題是 10:40:25。我們可以把給定學生的四題最晚提交時間由早到晚排序，然後把其對應的題號印出。以編號 19 的學生為例，依序是 4、1、2、3。

在本題中，我們將給定一至五個學生編號，請把這幾位學生的所有提交時間找出來，就每一題求出所有給定學生中的最晚提交時間以及這次提交的提交學生編號，把這四個最晚提交時間由早到晚排序，然後把對應的題號和學生編號印出。舉例來說，如果給定的學生編號是 17、19、20，則第一題到第四題的最晚提交時間依序是 12:17:02、12:20:01、12:06:24、11:59:17，分別是由編號 17、20、19、20 的學生提交的。將四題的最晚提交時間由早到晚排序後，依序是第四題、第三題、第一題、第二題。因此我們要依序印出的就是 4 與 20、3 與 19、1 與 17，最後則是 2 與 20。

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。在每個檔案中會有三行，第一行包含一個字串，代表需讀取之檔案的絕對路徑；第二行包含一個整數 n ，表示第三行有幾個學生編號。第三行包含 n 個整數，代表學生編號。兩個整數之間被一個空白隔開。我們只會給你隨本題附上的這個「midterm2.csv」檔案。

讀入資料後，請按照題目的規定印出四行，每一行有兩個數字，分別是一個題號和一個學生編號。第一行是最後提交時間最早的題號與做該次提交的學生編號，第二行是最後提交時間次早的題號與做該次提交的學生編號，依此類推。舉例來說，如果輸入是

```
C:/thisIsAnExamplePath/midterm2.csv
3
17 19 20
```

則輸出應該是

4 20

3 19

1 17

2 20

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。