商管程式設計（112-2）

作業六

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/`）爲第一、二、三題各上傳一份 Python 3.9 原始碼（以複製貼上原始碼的方式上傳）。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。

這份作業的截止時間是 ***4 月 27 日中午十二點***。在你開始前，請閱讀課本的第八、十一、十二、十四章[1]。爲這份作業設計測試資料並且提供解答的助教是秦孝媛。

第一題

（20 分）在一個 TXT 檔中記錄著某個連鎖零售商的歷史銷貨紀錄，每一列是一筆銷貨紀錄（第一列是欄位名稱除外），共有五個欄位，依序是門市編號（SID）、日期（DATE）、商品編號（PID）、購買數量（QTY），以及銷售當下該商品的單價（UNIT_PRICE）。example1.txt 是一個範例，其內容如下：

```
SID,DATE,PID,QTY,UNIT_PRICE
PD,2024/4/6,1,2,10
PD,2024/4/5,2,3,11
PB,2024/4/5,3,1,30
BF,2024/4/5,1,2,40
BAK,2024/4/4,3,2,15
BF,2024/4/4,4,1,33
BF,2024/4/3,5,1,10
BF,2024/4/3,5,1,10
```

在這個範例中，門市 PD 在 2024/4/6 的第一筆銷貨賣了 2 個商品 1，這一筆的營收共 20 元，而門市 BAK 在 2024/4/4 賣了 2 個商品 3，這一筆的營收共 30 元，依此類推。此外，請留意一個門市在一天內可能重複賣同一個商品複數次，例如 BF 在 2024/4/3 就賣了商品 5 兩次，每次都是賣了 1 個、營收 10 元。像這樣的兩筆交易會被分成兩筆記錄而非合併成一筆，以示那並不是一筆交易內賣掉 2 個商品 5。最後，即使是同一個門市在同一天賣同一個商品，只要是不同交易，就可能有不同的成交價格（例如某位客戶可能帶著折價券來消費）。理想上這樣的資料檔中應該也會有客戶編號，但爲了簡單起見，在本題中我們就忽略客戶編號。已知門市編號是長度爲 1 至 3 碼的大寫英文字母，商品編號、銷售數量與單價則都是正整數。

在本題中，你將被給定一個 TXT 檔以及一個指定的匯總方式：

- 如果指定的匯總方式是門市編號，請印出在整個 TXT 檔中總營收金額最高的前三個門市的門市編號及其總營收金額，依照總營收金額由高到低依序印出，金額相同則依照門市編號由小到大印出。如果在總營收金額的第三名附近平手，則在平手的門市中只印出編號最小的那幾位，以使最

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

終印出的只有三個門市的相關資訊。由於門市編號是字串，在幫門市編號比大小時，長度較短的編號被視爲比較小，如果長度相同，則第一個字母的字典順序排序較前（A 最前、B 次之，依此類推）的編號被視爲比較小，再平手則比第二個字母，再平手則比第三個字母。以前面的例子來說，門市編號由小到大依序是 BF、PB、PD、BAK。

- 如果指定的匯總方式是商品編號，則請印出總銷售筆數最高的前三個商品的產品編號及其總銷售筆數，依照總銷售筆數由高到低依序印出，銷售筆數相同則依照商品編號由小到大印出（按照數字順序）。如果在總銷售筆數的第三名附近平手，則在平手的產品中只印出編號最小的那幾個，以使最終印出的只有三個商品的相關資訊。

舉例來說，如果要以門市編號匯總，則此時門市 BF、PB、PD、BAK 的總營收金額依序是 133、30、53、30，因此要印出的三個門市的資訊依序是門市 BF 的 133 元、門市 PD 的 53 元，以及門市 PB 的 30 元（和門市 BAK 並列第三，此時挑門市編號小的）。如果要以商品編號匯總，則此時商品 1、2、3、4、5 的總銷售筆數依序是 2、1、2、1、2，因此要印出的三個商品的資訊依序是商品 1 的 2 筆、商品 3 的 2 筆，以及商品 5 的 2 筆。

注意：本題當然也有很多作法，大家可以任意挑自己喜歡的，也沒有助教會去看大家的程式碼。但相較於用 list 寫，用 dictionary 或其他資料結構顯然會比較好喔，建議大家還是趁這題做練習。

輸入輸出格式

系統會提供一共數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有 3 列，第一列是一個字串代表要讀入的 TXT 檔在 PDOGS 上的檔名（請直接把這個字串當成檔名做檔案讀取即可），第二列是此 TXT 檔的資料筆數 n（請注意 TXT 檔會有 $n+1$ 列，因爲有第一列的欄位名稱），第三列是一個英文大寫字元 S 或 P，S 表示以門市匯總，P 表示以商品匯總。TXT 檔中的第一欄是門市編號，是長度爲 1 至 3 碼的大寫英文字母；第二欄是日期，格式爲「yyyy/mm/dd」，年份是四位數字的西元年，月份和日期遇到只有個位數的時候不會補零；第三欄是商品編號，是一個介於 1 和 100 之間的整數（包含 1 和 100）；第四欄是購買數量，是介於 1 和 10 之間的整數（包含 1 和 10）；第五欄是單價，是介於 1 和 100 之間的整數（包含 1 和 100）。已知至少有 3 個門市和 3 個商品，且 $8 \leq n \leq 1000$。

請依題目指定的匯總方式，讀取 TXT 檔中的內容，並印出指定結果，印出時每一個門市或商品獨立一列，該列中先印出門市或商品編號，接著一個逗點，接著是該門市的總營收金額或該商品的總銷售筆數。舉例來說，如果輸入是

```
./assisting_data/example1.txt
8
S
```

則輸出應該是

```
BF,133
PD,53
PB,30
```

如果輸入是

```
./assisting_data/example1.txt
```

```
8
P
```

則輸出應該是

```
1,2
3,2
5,2
```

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 3.9 程式碼。當然，你應該寫適當的註解。針對這個題目，你***可以***使用上課沒有教過的方法。

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。

第二題

（20 分）承第一題，在本題中你依然會被給定一個 TXT 檔，格式也如第一題所述，但這次要請你寫一個程式，針對給定的門市編號與商品編號做搜尋。針對每個給定的輸入，如果它是個門市編號，請印出它在該 TXT 檔中的總營收金額；如果他是個商品編號，則印出它在該 TXT 檔中的總銷售筆數；如果該字串並非任何一個存在於該 TXT 檔的門市編號與欄位編號，則印出指定的錯誤訊息「BAD!!」。以第一題的例子來說，如果給定的輸入爲「BF」，請印出「133」；如果給定的輸入爲「4」，請印出「1」；如果給定的輸入爲「BD」或「6」，請印出「BAD!!」

在本題中，你需要按照規定定義並實作一個函數 `find_store_or_product`，該函數有三個參數 `id`、`store_rev` 和 `product_sales_cnt`，其形態依序爲字串、dictionary 和 dictionary，其中 `store_rev` 的 key 是門市編號（型態爲字串）、value 是該門市的總營收金額（型態爲整數），`product_sales_cnt` 則 key 是商品編號（型態爲整數）、value 是該商品的總銷售筆數（型態爲整數）。如果 `id` 存在於 `store_rev` 中，表示 `id` 是一個存在的門市編號，如此則回傳該門市的總營收金額；如果 `id` 被轉型成整數後存在於 `product_sales_cnt` 中，表示 `id` 是一個存在的商品編號，如此則回傳該商品的總銷售筆數；如果以上皆否，此時應從此函數拋出一個例外（exception）給呼叫此函數的地方，型態爲 `KeyError`。

在本題中，助教會在 PDOGS 上設置好正確地讀取 TXT 檔的內容、生成兩個 dictionary、呼叫你的函數、印出結果的主程式。你的程式會被大概這樣呼叫：

```
try:
    result = find_store_or_product('6', store_rev, product_sales_cnt)
    print(result)
except KeyError as e:
```

```
    print("BAD!!")
```

本題主要是讓大家認識、練習例外處理，雖然有強迫大家實作，但也沒有助教會去看大家的程式碼。如果理解 dictionary 和例外處理的原理和語法，這一題應該不是很難，但還是希望大家趁著寫作業的機會試著寫例外處理。「丟出例外」和「回傳一個特定數值」最不一樣的地方，是函數回傳值之後，不能保證呼叫此函數的地方會好好地根據回傳值做處理（甚至回傳值可能會完全被忽略），但丟出例外的話就一定要被處理（否則會 run-time error）。當有許多人一起寫一個程式（例如大家的期末專案時），這會是比較好的作法。此外，許多同學未來修課或就業時，會遇到的一個典型要寫程式的任務，是寫程式處理真實資料，而真實資料通常充滿了各式各樣的「例外」，如果能善加使用例外處理，就可以讓自己的程式碼看起來專業（進而獲得合作的夥伴的認可），實用上也比較不會出錯，更利於持續維護、擴充。總之，請務必練習看看吧！

特別說明：助教已經在 PDOGS 上放好會去呼叫這個函數的主程式碼了，所以大家只需要實做並且上傳這個函數的定義程式碼就好，PDOGS 會把助教寫的主程式跟你寫的函數拼在一起直譯、執行。換個角度講，大家也只能上傳這個函數，如果上傳了其他程式碼（例如你自己寫的主程式），PDOGS 就會因此拼出重複的程式碼，結果反而就會無法直譯跟執行了。總之，你上傳的程式碼應該只包含這個函數的定義。

輸入輸出格式

系統會提供一共數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有 $m+2$ 列，第一列是一個字串代表要讀入的 TXT 檔在 PDOGS 上的檔名（請直接把這個字串當成檔名做檔案讀取即可），第二列是此 TXT 檔的資料筆數 n（請注意 TXT 檔會有 $n+1$ 列，因爲有第一列的欄位名稱），一個逗點，和後續要被查詢的字串個數 m，第三列起的 m 列每一列是一個字串，長度爲 1 到 10 個字元（包含 1 跟 10），只含有大寫英文字母和數字，代表要被查詢的可能是門市編號或商品編號也可能什麼都不是的字串。已知至少有 3 個門市和 3 個商品、$8 \le n \le 1000$、$1 \le m \le 10$。TXT 檔的格式和第一題一模一樣。

請依題目規定，針對每個被查詢的字串印出查詢結果及一個換行字元。舉例來說，如果輸入是

```
./assisting_data/example1.txt
8,4
BF
BD
4
6
```

則輸出應該是

```
133
BAD!!
1
BAD!!
```

如果輸入是

```
./assisting_data/example1.txt
8,4
BF1
BDD
4
66666
```

則輸出應該是

```
BAD!!
BAD!!
1
BAD!!
```

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僅包含題目指定的函數。當然，你應該寫適當的註解。針對這個題目，你***可以***使用上課沒有教過的方法。

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。

第三題

（60 分）在作業五第三題中，我們寫了程式去處理文章和句子，並且透過公式去判斷正負向情緒。若第 i 句話的正向字次數爲 x_i^{pos}，負向字次數爲 x_i^{neg}，則這個句子的情緒分數 y_i 爲

$$y_i = x_i^{\text{pos}} - x_i^{\text{neg}}。$$

這個公式雖然有點道理，但也有點沒道理，特別是公式中每個正向詞和負向詞的權重（weight）都是一樣的，不免顯得奇怪，畢竟「老師教得不太好」跟「老師教得糟透了」跟「我從未見過如此糟糕的老師，完全不適任」看起來是有情緒差異。

當然，要我們現在就去判斷這些細微的情緒差異，是有點太勉強了，也遠超出這門課的範疇；在本題我們就只探討一個議題，就是「正向詞和負向詞彼此之間的權重應該是多少」。更具體地說，我們要把我們的公式改成

$$y_i = \alpha x_i^{\text{pos}} - \beta x_i^{\text{neg}}，$$

其中 α 和 β 是這個公式的參數，給定不同的參數值，這個公式就會幫每一句話算出不同的分數。我們可以用很簡單的方式幫每一句話分類成「正向」或「負向」：如果分數 y_i 大於等於零就是正向，小於零就是負向。那麼只要決定了 α 和 β，我們就會得到一個非常簡單的分類模型（classification model），可以用於判斷一句話或一篇文章的情緒傾向了！

像 α 和 β 這樣的量，在機器學習的世界裡則被稱爲一個模型的超參數（hyper-parameter）。顯然超參數的值會影響模型的行爲和表現，而研究者（資料科學家）的任務（之一）就是透過歷史資料，尋找最合適的超參數。在我們這個「監督式學習」的分類問題中，我們需要有「有標籤」（labeled）的歷史資料，亦即我們需要一堆句子，且已經有足夠可信的人去閱讀這些句子並且標註其情緒傾向了（例如每個句子我們都給五個人標註，讓五個人都去標註情緒正負向，最終以多數決），然後透過程式或演算法去找能讓模型表現最好的超參數。

在本題中，就讓我們來稍稍地體驗一下這樣的流程吧！你將被給定 n 個句子（字串）、它們被標註爲正向或負向的標註結果，以及一個情緒辭典[2]。情緒辭典包含 5 個正向字「good」、「best」、「awesome」、「excellent」、「wonderful」，以及 4 個負向字「bad」、「worst」、「stupid」、「shame」。你的任務是搜尋 α 和 β 這兩個超參數，看看哪一組超參數能在這 n 個句子得到最正確的判斷。爲了簡單起見，在本題中你只需要考慮 (α, β) 爲 $(1,1)$、$(1,2)$、$(1,3)$、$(2,1)$、$(2,3)$、$(3,1)$、$(3,2)$ 這七種組合，看看哪一種能得到最多的判斷正確即可。

我們用表 1、表 2 和表 3 的範例做更仔細的說明。表 1 中有 7 個句子以及已知的標註結果。請注意標註結果跟我們等等會數的情緒詞個數沒有關係；標註結果是直接被給定的、當作標準答案使用的。

i	句子	情緒標註
1	The food was good AND excellent. However, the service was bad!	負
2	The movie was awesome; the experience was wonderful.	正
3	The instructor tried his best, but the course IS still the worst.	負
4	I cannot tell whether the food iS good Or bad. But the waiter IS stupid.	負
5	The laptop IS the best! Wonderful performance! Awesome!	正
6	This book makes no sense; I feel shame that the authors are NTU alumni.	負
7	Winning the award was an “terrific” experience, truly the best-feeling.	正

表 1: 範例訓練資料

在表 2 中，第二、三欄是各個句子裡含有的正向詞與負向詞個數，後面七欄則是在各種 (α, β) 的組合中會幫每個句子計算而得的情緒分數。舉例來說，第一個句子有 2 個正向詞和 1 個負向詞，用 $(\alpha, \beta) = (1,1)$ 時算出的情緒分數爲 $1 \times 2 - 1 \times 1 = 1$，用 $(\alpha, \beta) = (1,2)$ 時算出的則爲 $1 \times 2 - 2 \times 1 = 0$，依此類推。

從表 2 的結果我們可以直接得到表 3，例如第一個句子用 $(\alpha, \beta) = (1,1)$ 時算出的情緒分數爲 $1 > 0$，因此模型判定爲正向；用 $(\alpha, \beta) = (1,2)$ 時算出的則爲 $0 \geq 0$，模型也判定爲正向，但用 $(\alpha, \beta) = (1,3)$ 時算出的則爲 $-1 < 0$，因此模型判定爲負向。將表 3 的模型判定結果與表 1 的人工標註結果（我們用此當作正確答案）比較，即可得到表 3 最後一列的判斷正確次數。在這個例子中，我們會選擇 $(\alpha, \beta) = (1,3)$，因爲這組超參數在訓練資料中有最高的判斷正確次數。

請寫一個程式，按照上面的指定流程，利用給定的訓練資料（句子及標註），在 (α, β) 的指定範圍內訓練出你能得到的最好的分類模型。情緒辭典的內容和比對的規則（大小寫不論，要恰好相同）都如作業五第三題所述。如果有複數組 (α, β) 都能得到同樣最高的判斷正確次數，就選 α 比較小的；如果還是平手，就選 β 比較小的。

特別說明：即使不考慮艱難的、進階的文意判讀，真實的訓練流程也會比本題所呈現的複雜許多。舉例來說，實務上可能不是把句子的情緒分成兩類（正向、負向），而是分成三類（加入「中性」）；可

[2]實務上的情緒辭典都非常地長，裡面的詞成千上萬，但這題我們簡單就好。

i	x_i^{pos}	x_i^{neg}	(α,β)						
			$(1,1)$	$(1,2)$	$(1,3)$	$(2,1)$	$(2,3)$	$(3,1)$	$(3,2)$
1	2	1	1	0	-1	3	1	5	4
2	2	0	2	2	2	4	4	6	6
3	1	1	0	-1	-2	1	-1	2	1
4	1	2	-1	-3	-5	0	-4	1	-1
5	3	0	3	3	3	6	6	9	9
6	0	1	-1	-2	-3	-1	-3	-1	-2
7	1	0	1	1	1	2	2	3	3

表 2: 範例情緒分數計算過程

i	(α,β)						
	$(1,1)$	$(1,2)$	$(1,3)$	$(2,1)$	$(2,3)$	$(3,1)$	$(3,2)$
1	正	正	負	正	正	正	正
2	正	正	正	正	正	正	正
3	正	負	負	正	負	正	正
4	負	負	負	正	負	正	負
5	正	正	正	正	正	正	正
6	負	負	負	負	負	負	負
7	正	正	正	正	正	正	正
判斷正確次數	5	6	7	4	6	4	5

表 3: 範例判斷結果與正確次數

能在五個人標註時，三比二或二比三的資料會略去不用；可能需要檢視每個標註者的標註品質，看有沒有人亂標；情緒字典或辭典的品質和完整性顯然也需要研究。此外，本題因爲只是體驗一下，所以設定了高度限定的超參數範圍，只要一一嘗試就能找到最佳超參數，但實務上搜尋最佳超參數通常需要求解複雜的最佳化問題，也需要特別設計的演算法；最後，許多的訓練過程需要把訓練資料進一步切成「訓練資料」跟「驗證資料」，但在本題我們也略過這個步驟。大家如果想進一步了解機器學習的基本原理與流程，在 Coursera 的第三門課中有一週會做相關介紹；在本題就按照題目敘述完成任務就好！

輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中將有 $n+2$ 列，第一列有一個整數 n；第二列至第 $n+1$ 列有 1 個英文句子字串，格式與作業五第三題一模一樣，字串內容中只包含大小寫英文字母、英文標點符號與空白字元。本題會出現的標點符號只有句點、逗點、冒號、分號、問號、驚嘆號、單引號、雙引號跟連接號（`.,:;?!'"-`）。給定的句子不一定符合正確的英文文法，標點符號前後也不一定有被正確使用的空白字元；句子開頭可能是空白字元；句子結尾也可能是空白字元。第 $n+2$ 列有 n 個整數，其中第 i 個整數代表第 i 個句子的標註結果，若爲 1 表示標註爲正向，若爲 0 表示標註爲負向，兩兩之間以一個逗點隔開。已知 $1 \le n \le 100$，每個句子的字元數不超過 1000。

請讀入以上資料，接著輸出 3 個整數，依序爲最佳的 α、最佳的 β，以及這組 (α, β) 的模型判斷正確次數，兩兩之間以一個逗點隔開。舉例來說，如果輸入是

```
7
The food was good AND excellent. However, the service was bad!
The movie was awesome; the experience was wonderful.
The instructor tried his best, but the course IS still the worst.
I cannot tell whether the food iS good Or bad. But the waiter IS stupid.
The laptop IS the best! Wonderful performance! Awesome!
This book makes no sense; I feel shame that the authors are NTU alumni.
Winning the award was an "terrific" experience, truly the best-feeling.
0,1,0,0,1,0,1
```

則輸出應該是

```
1,3,7
```

如果在訓練資料中第一個句子被標註爲正向，因此輸入是

```
7
The food was good AND excellent. However, the service was bad!
The movie was awesome; the experience was wonderful.
The instructor tried his best, but the course IS still the worst.
I cannot tell whether the food iS good Or bad. But the waiter IS stupid.
The lapTOP isss the BeSt  ! Wonderful   performance! Awesome!
This book makes no sense; I feel shame that the authors are NTUalumni.
Winning the award was an "terrific " experience ,truly the best-feeling.
1,1,0,0,1,0,1
```

則七個 (α, β) 組合的判斷正確次數依序（如表 3 中由左至右的順序）爲 6、7、6、5、7、5、6，$(1, 2)$ 和 $(2, 3)$ 都能得到最高的判斷正確次數，因此按照規則選擇 α 較小的 $(1, 2)$。輸出應該是

```
1,2,7
```

請注意在這個例子的第五句出現了很多奇怪的大小寫和不合理的連續空白，但這不影響我們判定正負情緒字的出現；第六句有出現算是錯字的「NTUalumni」（應該要是「NTU alumni」），但也不影響我們判斷；第七句的右雙引號左邊多了一個空白字元，逗點前面多了一個後面少了一個空白字元，但這都不影響我們判定。最後，第七句的「best-feeling」中是含有情緒字「best」的，因爲「best」的前後各一個字元都不是英文字母。

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 3.9 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之前作業說過可以使用的語法、跟 set、tuple、dictionary、datetime、例外處理、檔案讀取等有關的所有操作與 Python 內建函數。

- 沒有確定不可以使用的語法，但沒教過的不能用就是了。

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 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。本題共有 20 組測試資料，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可讀性（包含排版、變數命名、註解等等）。請寫一個「好」的程式吧！