資料庫管理（110-1）
作業一

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/`）為第一題上傳一份 C++ 17 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 ***2021 年 9 月 29 日早上八點***。

特別說明：當你耐心地閱讀完這份作業後，會發現這是一個「寫程式處理資料」的任務。C++ 是個很棒的程式語言，但說到處理資料，通常使用 Python 等其他程式語言似乎更為合理。若你想知道為什麼這份作業要求 C++，原因如下：

1. 這門課照資管系的傳統，是要求先修過「作業系統」的，因為若想確實瞭解資料庫管理系統的運作原理，就不能完全不瞭解作業系統如何管理硬碟、記憶體以及其他相關知識。當然就現況來說，校內的資料庫相關課程開得並不多，但想學習的學生卻不少，因此我暫時也沒有打算強制要求所有學生都修過作業系統。但即使如此，修我們這門課的學生還是要對硬碟、記憶體有夠多的認識，更具體地來說，必須要會寫 C 或 C++ 的指標去「管理」記憶體，所以 C 或 C++ 的能力還是得要有的，課程中也確實會用到。
2. （宣稱）想修這門課的人實在太多了，在 9 月 19 日的當下，有約 60 人選上以及約 160 人登記想加選。就算登記想加選的只有一半是真心的，總人數依然遠超過我們的負荷量（大約 100 人）。既然如此，多一個篩選門檻也是好的，畢竟會 C 和 C++ 的同學修這門課，收穫確實會比較大。
3. 會 C 或 C++ 都很好，但因為資管系是教 C++，所以我們的官方要求就是 C++。如果你擅長的是 C，那倒是無所謂，寫出來的程式碼讓 C++ 17 的編譯器能正確編譯就好。

總之，不要擔心，老師不是傻到覺得 C++ 比 Python 更適合處理資料。第一份作業有兩個目的，第一是讓大家知道我們是認真要求學生會寫 C 或 C++ 的，若你不會，建議還是不要勉強；第二是讓大家感受一下處理資料是件麻煩事。隨著課程進行，希望我們能讓你相信，相較於把資料裝在檔案裡並且直接存取資料，把資料裝在資料庫裡有很多好處。

第一題

（100 分）某公司是近年來在線上教育領域嶄露頭角的臺灣新創公司，其自行開發的產品將學習融入線上遊戲，在中小學教育、企業內訓等領域都頗受好評。

為了因應素養教育的興起，集團在 2019 年成立了子公司，專門投入中小學學童的素養教育。其商業模式為訂閱制，學生家長訂閱並繳交月費後，便可為其子女（或任意指定對象）開通學員帳號，學員即可在線上遊戲中回答任務問題、賺取金幣、擴張領土，達到寓教於樂的效果。在遊戲中，付費訂閱的玩家可以獲得「素養任務」，也可以在針對素養任務的題目作答後閱讀參考解答，而每位玩家針對每份素養任務的每個題目的每次作答，都會被系統記錄下來，供經營團隊瞭解每位玩家的學習活動與情況，進而在合適的時機對每位玩家給予不同的關懷、挑戰，或根據這些記錄優化系統、修正題目難易度或更改介面設計。

若要給予每位玩家不同的關懷或挑戰，就必須先將玩家分組[1]。在本題中，我們將分析每位玩家相關的任務作答記錄，並透過一些指標來將玩家區分為幾個類別。以下我們介紹（簡化後的）遊戲進行方式與資料記錄方式。

資料說明

在這款遊戲中，系統會定期派給玩家任務，玩家可以透過回答任務題目來賺取金幣、擴張領地等。資料中會記錄每個任務被派給每位玩家的時間點，而同一份任務被派給不同玩家的時間點可能有些微差異。玩家看到系統派的任務後，可以在任何時候接取這個任務，接取任務後才能開始（但不一定要立刻）進行任務，一個任務包含了多個題目，玩家必須至少回答其中一個題目才能「交卷」來完成該任務（也就是有些題目可以不作答），系統會依據答題花費時間與答對題數給玩家對應獎勵。另外，玩家可以在任何時候接取多個任務，也可以在任何時候暫停正在進行中的任務。

在附上的「user_missions.csv」檔中，我們有從 2019/03/30 到 2021/05/09 在閱讀優中每位玩家在遊戲中的任務作答記錄，檔案中共有 441,562 列，每一列代表一位玩家對一個任務的作答記錄，各欄位說明如下：

- id：每一筆任務作答記錄的 id，為一個字串。
- user_id：每一位玩家的 id，為一個字串。
- mission_id：每一個任務的 id，為一個字串。
- status：玩家對任務的狀態。「已派發 Assigned」代表系統已將該任務派給玩家，但玩家尚未接取該任務；「進行中 Ongoing」代表玩家已接取該任務但尚未完成；「已暫停 Paused」代表玩家接取該任務後，在任務尚未完成前主動暫停任務；「已完成 Completed」代表玩家已接取該任務，並完成該任務。
- correct_count：玩家完成該任務時的答對題數。
- answered_count：玩家完成任務時的作答題數，此數值必大於或等於 1，但可能不等於該任務中的題目總數量。
- answer_duration：玩家回答該任務之問題所花的時間，單位為秒。
- started_at：一個格式為 yyyy-mm-dd hh:mm:ss 的字串，代表玩家接取該任務的時間點，包含日期與時間。
- finished_at：一個格式為 yyyy-mm-dd hh:mm:ss 的字串，代表玩家完成該任務的時間點，包含日期與時間。
- created_at：一個格式為 yyyy-mm-dd hh:mm:ss 的字串，代表該任務被派給玩家的時間點，包含日期與時間。

要注意的是，answer_duration 的數值不一定等於 started_at 與 finished_at 的時間差，因為玩家接取任務後不一定會立刻進行任務，而 answer_duration 只單純記錄答題花費時間。另外，資料中的一些欄位可能出現空值的情形，取決於玩家對任務的狀態（status），對應關係如表 1 所示。

[1]學術上可能會稱呼這個動作為「分群」（clustering），以與「分類」（classification）做區隔。本題由於不牽涉到機器學習、預測等議題，分組、分群或分類精神上都是一樣的，因此在敘述上我們也可能將這些詞彙混合使用。

status	correct_count	answered_count	answer_duration	started_at	finished_at	created_at
Assigned	空值	空值	空值	空值	空值	非空值
Ongoing	空值	空值	空值	非空值	空值	非空值
Paused	空值	空值	空值	非空值	空值	非空值
Completed	非空值	非空值	非空值	非空值	非空值	非空值

表 1: 資料空值說明

資料前處理與指標計算

給定這個檔案與一個指定的日期，我們想要研究在這個日期前後玩家的作答記錄情形。我們會透過以下步驟將資料進行前處理與指標計算，說明如下：

1. 篩選出那些有完整作答記錄的資料。可以透過篩選 status 為「Completed」的那些資料，或去除 answered_count 為空值的那些資料來完成。

2. 針對這個指定日期，再篩選出所有 started_at 介於該日期前四週的資料（不包含該日期，共 28 天），例如若指定日期為 2021-02-22，則需要篩選那些 started_at 介於（頭尾都有包含）2021-01-25 至 2021-02-21 的那些資料。

3. 針對那些有出現在篩選後資料的玩家，對於 started_at 介於指定日期前一週、前兩週、前三週、前四週的資料，分別計算每週分別的總作答題數、總答對題數、作答正確率。其中，總作答題數、總答對題數為該週所有對應數值的累加，作答正確率的計算公式為「該週總答對題數除以該週總作答題數」，並四捨五入至小數點後第三位（不需轉換為百分數）。請注意如果某位玩家只在這四週中的某幾週有記錄，仍需計算全部四週的所有指標，並在沒有記錄的週次留下空值，下一步驟會說明如何填補這些空值。

4. 若某位玩家在某幾週無任何作答資料，則將那幾週的總作答題數、總答對題數都補上 0，作答正確率補上有資料的週次之作答正確率取算數平均，並四捨五入至小數點後第三位（不需轉換為百分數）。表 2 為作答正確率補值的範例。

週次	前四週	前三週	前二週	前一週
作答正確率補值前	空值	0.7	0.5	空值
作答正確率補值後	0.6	0.7	0.5	0.6

表 2: 作答正確率補值範例

5. 對於玩家在這四週間的作答總題數變化，計算出三個對應斜率。假設該玩家在前一週、前兩週、前三週、前四週（補值後的）作答總題數分別為 a_1、a_2、a_3、a_4，並將它們以座標上的四個點 $(-1, a_1)$、$(-2, a_2)$、$(-3, a_3)$、$(-4, a_4)$ 表示，接著依序計算三個指標：

   - 前四週至前一週的作答總題數變化趨勢：$(-1, a_1)$、$(-2, a_2)$、$(-3, a_3)$、$(-4, a_4)$ 四個點的迴歸直線斜率[2]。
   - 前三週至前一週的作答總題數變化趨勢：$(-1, a_1)$、$(-2, a_2)$、$(-3, a_3)$ 三個點的迴歸直線斜率。

[2]若有 n 筆數據 (x_i, y_i)，$i = 1, ..., n$，且設 $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$、$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$，則其迴歸直線斜率為 $\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$。

- 前兩週至前一週的作答總題數變化趨勢：$(-1, a_1)$、$(-2, a_2)$ 兩個點的迴歸直線斜率。

最後，請將這三個斜率四捨五入至小數點後三位。

你可能已經注意到了，如果只要完成以上的資料前處理與指標計算，資料中 id、answer_duration、finished_at、created_at 這四個欄位是不會用到的。

資料處理範例

以下用一個範例逐步說明資料處理的過程。假設指定的日期為 2021-02-22，且在資料中，有某位 user_id 為 001 的玩家的任務作答記錄如表 3 所示[3]。

id	user_id	mission_id	status	correct_count	answered_count	started_at
1	001	101	Completed	4	5	2021-01-24 14:00:00
2	001	102	Completed	4	10	2021-01-25 20:00:00
3	001	103	Completed	1	2	2021-01-31 21:00:00
4	001	104	Completed	2	5	2021-02-09 11:00:00
5	001	105	Completed	3	5	2021-02-11 12:00:00
6	001	106	Completed	15	15	2021-02-20 22:00:00
7	001	108	Ongoing	–	–	2021-02-21 20:00:00
8	001	107	Completed	0	5	2021-02-22 20:00:00
9	001	109	Paused	–	–	2021-02-28 20:00:00
10	001	110	Assigned	–	–	–

表 3: user_id 為 001 的用戶的任務作答記錄

若我們執行前述的五個資料處理步驟，依序會得到以下結果：

1. 篩選出那些有完整作答記錄的資料。我們會刪除 id 為 7、9、10 的那三筆資料，篩選後的結果如表 4 所示。

id	user_id	mission_id	status	correct_count	answered_count	started_at
1	001	101	Completed	4	5	2021-01-24 14:00:00
2	001	102	Completed	4	10	2021-01-25 20:00:00
3	001	103	Completed	1	2	2021-01-31 21:00:00
4	001	104	Completed	2	5	2021-02-09 11:00:00
5	001	105	Completed	3	5	2021-02-11 12:00:00
6	001	106	Completed	15	15	2021-02-20 22:00:00
8	001	107	Completed	0	5	2021-02-22 20:00:00

表 4: 資料處理第一步驟後的結果

2. 篩選出所有 started_at 介於指定日期前四週的資料。由於指定日期為 2021-02-22，因此需要篩選出 started_at 介於（頭尾都有包含）2021-01-25 至 2021-02-21 的那些資料。我們會刪除 id 為 1、8 的那兩筆資料，篩選後的結果如表 5 所示。

[3]為了版面美觀，沒有列出 answer_duration、finished_at、created_at 這三個不會用到的欄位。

id	user_id	mission_id	status	correct_count	answered_count	started_at
2	001	102	Completed	4	10	2021-01-25 20:00:00
3	001	103	Completed	1	2	2021-01-31 21:00:00
4	001	104	Completed	2	5	2021-02-09 11:00:00
5	001	105	Completed	3	5	2021-02-11 12:00:00
6	001	106	Completed	15	15	2021-02-20 22:00:00

表 5: 資料處理第二步驟後的結果

3. 對於 started_at 介於指定日期前一週、前兩週、前三週、前四週的資料[4]，分別計算每週分別的總作答題數、總答對題數、作答正確率。結果如表 6 所示，請注意在「前三週」時這位玩家沒有任何資料，因此先留下空值。另外，你可以將計算後的資料以任何格式儲存，以下的範例是將每個玩家的所有相關資料儲存成四列。

user_id	前幾週	總答對題數	總作答題數	作答正確率
001	前一週	15	15	1.000
	前兩週	5	10	0.500
	前三週	–	–	–
	前四週	5	12	0.417

表 6: 資料處理第三步驟後的結果

4. 若某位玩家在某幾週無任何作答資料，則將那幾週的總作答題數、總答對題數都補上 0，作答正確率補上有資料的週之作答正確率取算數平均。補值後結果如表 7 所示，其中作答正確率補上的數值為 $\frac{1.000+0.500+0.417}{3} = 0.639$。

user_id	前幾週	總答對題數	總作答題數	作答正確率
001	前一週	15	15	1.000
	前兩週	5	10	0.500
	前三週	0	0	0.639
	前四週	5	12	0.417

表 7: 資料處理第四步驟後的結果

5. 對於玩家在這四週間的作答總題數變化，計算出三個對應斜率。

   - 代表前四週至前一週的作答總題數變化趨勢：$(-1, 15)$、$(-2, 10)$、$(-3, 0)$、$(-4, 12)$ 四個點的迴歸直線斜率，數值為 1.900。
   - 代表前三週至前一週的作答總題數變化趨勢：$(-1, 15)$、$(-2, 10)$、$(-3, 0)$ 三個點的迴歸直線斜率，數值為 7.500。
   - 代表前兩週至前一週的作答總題數變化趨勢：$(-1, 15)$、$(-2, 10)$ 兩個點的迴歸直線斜率，數值為 5.000 。

[4]在這個範例中，指定的日期為 2021-02-22，因此它的前一週是 2021-02-15 至 2021-02-21、前兩週是 2021-02-08 至 2021-02-14、前三週是 2021-02-01 至 2021-02-07、前四週是 2021-01-25 至 2021-01-31。

玩家分類方式

完成資料前處理與指標計算後，我們可以得到每位玩家在指定日期前四週的 $(3 \times 4) + 3 = 15$ 個指標，包含這四週各自的總作答題數、總答對題數、作答正確率，再加上三個斜率，且每一個指標都不應該是空值。接著我們會進一步利用這些指標判斷玩家的「活躍程度」，將玩家分成四個類型。

對於每一位玩家來說，令 s 是這四週總作答題數的平均，x 是三個斜率中有幾個是非負的，$x \in \{0, 1, 2, 3\}$。給定一個平均作答總題數門檻 t，我們可以依照以下標準將玩家分為四個類型：

- 類型 1 玩家：$s \geq t$ 且 $x \geq 2$。
- 類型 2 玩家：$s \geq t$ 且 $x < 2$。
- 類型 3 玩家：$s < t$ 且 $x \geq 2$。
- 類型 4 玩家：$s < t$ 且 $x < 2$。

請計算四個類型各自的玩家人數，與該類型中玩家的平均作答正確率。平均作答正確率的計算方式說明如下：

1. 先計算每位玩家這四週作答正確率的平均（以「四週平均作答正確率」稱呼），並四捨五入至小數點後第三位（不需轉換為百分數）。
2. 對每個類型中所有玩家的「四週平均作答正確率」再取算術平均，並轉換為百分數後，無條件捨去至整數位。如果有某一類型的玩家人數為 0，請將這類型中玩家的平均作答正確率直接設為 0%。

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，輸入資料共有 3 行，第一行包含一個字串，代表需讀取檔案之絕對路徑[5]；第二行為一個 yyyy-mm-dd 字串，代表指定的日期；第三行為一個非負整數 t，代表平均作答總題數門檻。

讀入資料後，請依照題目的規定篩選出指定日期前四週的資料，並計算出各類型的玩家人數，以及該類型中玩家的平均正確率。輸出共有四行，依序為類型 1、類型 2、類型 3、類型 4 的資訊，每一行為該類型的人數與平均正確率（轉換為百分數後無條件捨去至整數位），兩個數字之間用一個半形逗號隔開。

舉例來說，如果輸入是

```
C:/ThisIsAnExamplePath/user_missions.csv
2021-02-22
15
```

則輸出應該是

```
266,54
153,52
```

[5]在批改系統中，我們會把提供給你的 user_missions.csv 檔案放在輸入資料第一行的字串所代表的位置。換言之，請直接使用輸入資料第一行的字串做你的程式讀取檔案時的檔案路徑字串。

```
262,52
143,50
```

如果輸入是

```
C:/ThisIsAnExamplePath/user_missions.csv
2021-02-22
30
```

則輸出應該是

```
36,53
48,51
492,53
248,51
```

如果輸入是

```
C:/ThisIsAnExamplePath/user_missions.csv
2021-01-01
15
```

則輸出應該是

```
196,51
165,50
271,55
152,53
```

如果輸入是

```
C:/ThisIsAnExamplePath/user_missions.csv
2021-01-01
30
```

則輸出應該是

```
25,45
20,50
442,53
297,51
```

如果輸入是

```
C:/ThisIsAnExamplePath/user_missions.csv
2020-10-01
2
```

則輸出應該是

```
477,50
290,50
71,60
14,73
```

如果輸入是

```
C:/ThisIsAnExamplePath/user_missions.csv
2020-10-01
100
```

則輸出應該是

```
0,0
0,0
548,51
304,51
```

如果輸入是

```
C:/ThisIsAnExamplePath/user_missions_another.csv
2021-02-22
15
```

則輸出應該是

```
273,55
127,52
259,51
156,50
```

特別說明：大家獲得這份題目的同時，也可以下載這邊的範例輸入輸出所使用的「user_mission.csv」和「user_mission_another.csv」。只要把這兩個 CSV 檔放在你的 C++ 程式找得到的地方，那麼若你的程式是正確的，就應該能在讀入指定檔案和相對應的參數後，得到題目上顯示的正確結果。在 PDOGS 上我們也有設定自動批改的範例輸入輸出（sample），繳交程式碼後 PDOGS 也會檢視你的程式針對 sample 是否能算出正確結果，但 PDOGS 批改時使用的 CSV 檔，未必就是提供給大家下載的這兩個 CSV 檔喔！

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以及程式運行是否滿足時間和記憶體限制。一筆測試資料佔 5 分。

我們鼓勵同學間互相討論，但抄襲是嚴格禁止的。在本課程中，我們使用國際上被許多課程使用的程式比對系統，若兩份原始碼被判定為過於雷同，第一次我們會給予警告，第二次之後將認定為抄襲並且將該份作業以零分計。如果你想跟同學互相討論，請僅止於想法、解題策略上的討論，但程式碼仍應由你親手撰寫。

在期限前，你可以做無限次上傳，上傳次數不影響得分，但為了避免浪費系統資源，請自行做好檢查再上傳，不要用 PDOGS 當你的 compiler。請注意，我們是用截止前最後一次上傳的程式計分，不適用最高分的那一次上傳。