

資料結構與進階程式設計 (106-2)

程式作業二

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++ 原始碼 (以複製貼上原始碼的方式上傳)，為第二題上傳一份 PDF 檔，為第三題做同儕互評。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。

這份作業的前兩題的截止時間是 **2018 年 4 月 16 日下午一點**；第三題的截止時間是 **2018 年 4 月 20 日下午一點**。在你開始前，請閱讀第一本課本的第 12、13、14、16 章¹ 以及第二本課本的第 3、4 章² 為這份作業設計測試資料並且提供解答的是兩位助教。

第一題

(60 分) 我們上課時已經介紹過 Ride sharing (共乘) 系統了，現在就讓我們來實做一個吧！

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，也**不可以**使用 `<vector>` 裡面的東西。

除此之外，你**應該**利用封裝 (encapsulation)、繼承 (inheritance)、多型 (polymorphism)、樣板 (template)、例外處理 (exception handling) 去把程式寫得更好。更重要地，你**應該**利用 bag 資料結構來裝車子和乘客，並且練習 link-based bag。若你沒有這麼做，第三題同儕互評時可能會被扣分。

事實上，上述進階程式設計方法都可以協助你在開發程式時避免不必要的心力耗費，更能減少開發過程的錯誤，希望大家能盡量思考怎樣使用、為什麼要使用這些方法。此外，由於車子和乘客的上線數量會一直變動，而且你不會想要在搜尋車子的時候把所有上線、離線的車子都搜尋一遍 (你應該只搜尋空車)，不然你的程式很可能無法通過 PDOGS 上的時間限制，所以你最好的作法本來就是實做 link-based bag。請大家試試看吧！

題目敘述

在本系統中會有以下兩種角色：車子和乘客。雖然應該還有司機的角色，但我們假設司機跟車子有一對一關係，所以就簡化為只有車子和乘客就好。就像 Uber 也有 UberBlack 這種高檔車可以選，我們的車子也分為兩種：普通車跟高檔車。普通車跟高檔車是完全不同的，亦即只有特定廠牌的車子才能被系統記錄為高檔車，其他車子就都是普通車。總之一輛車被加入系統時，系統就會指定它為高檔車或普通車，之後也不會改變。

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

²課本是 Carrano and Henry 著的 *Data Abstraction and Problem Solving with C++: Walls and Mirrors* 第六版。

不論是車子還是乘客，在系統中的狀態都有上線（online）和離線（offline）之分。上線中的車子分為空車（empty）與服務中（in-service）兩種，其中服務中包含已被指定給某個乘客、正在前往的路上，此時雖然車上沒有乘客，但我們依然說車子在服務中的狀態，且已經跟該乘客連結了。上線的乘客分為等待中（waiting）和服務中（in-service）兩種，其中等待中表示已被指派但還沒上車，而被服務中則表示正在前往目的地。這兩個狀態中此乘客都會有已經連結的車子。乘客上線叫車時若一時之間叫不到車，他會立刻離線。

乘客搭車是要付錢的，而車子的計價方式是由單趟基本價格加上依里程而計的價格。普通車和高檔車的價格是不同的，因此普通車有兩個價格參數（基本價格 a_R 和單位里程價格 b_R ），高檔車也有這兩個價格參數 a_L 和 b_L 且各比相對應的普通車參數高。如果乘客搭乘普通車走了 x 公里，則車資為 $a_R + b_R x$ ，若搭乘高檔車則為 $a_L + b_L x$ 。舉例來說，若普通車基本價格為 50 而每公里要 3 元，則搭 10 公里要付 80 元。請注意只要是普通車，都用一樣的計價公式；只要是高檔車，也都用一樣的計價公式。

在真實世界的共乘系統中，通常乘客和車子間會做雙向評價，也就是乘客和車子都會有某種形式的累積評分。在本題中，我們只考慮車子獲得的評分。我們假設每一趟服務結束後，該乘客都可以給予該車子 1 至 5 分的一個整數分數，這會被累加進車子的被評分次數與生涯累積總分。乘客當然也可以不給車子評分，那該車子的被評分次數與生涯累積總分就不會改變。在本題中，車子的被評分次數與生涯累積總分會影響其被平臺指派給乘客的順序，但平均分數則不會，所以你可以不用費心計算和保存這個數值。

另一個會影響車子和乘客配對的是屬性標籤。系統中有 n 種屬性標籤，像是「愛聊天」、「喜歡棒球」、「車上有小電視」之類的。每個車子或乘客都可以在數種喜好標籤中挑幾個貼在自己身上，這會影響到配對的順序。當一個乘客搜尋車子時，如果周圍有數輛空車，則和乘客有較多相同之屬性標籤的空車會較優先地被配對。

綜合以上，讓我們來說明平臺將車子指派給乘客的規則。乘客上線叫車時，會選擇他要找高檔車還是普通車，此時平臺就只搜尋此乘客選擇的車種。若乘客因為找不到偏好的車種而改搜尋另一種，我們想像成這個乘客是先離線再上線做搜尋。系統中定義有指派車子的最大距離 K ，乘客叫車時與乘客距離大於 K 公里的車子就不納入考慮。若乘客周圍 K 公里內只有一輛空車，平臺就會指定這輛車給乘客，讓乘客進入等待狀態；若沒有空車，則乘客會立刻離線（也許會再上線搜尋另一種車種）；若有複數輛空車，則每輛空車對此乘客的適配度分數是

$$s = r - kc + ht - pd,$$

其中 r 是該車的生涯累積總分、 c 是生涯總被評分次數、 t 是該車與該乘客共同的標籤個數、 d 是該車與該乘客間的距離（以公里計），而 k 、 h 與 p 則是系統參數。平臺將指派適配度最高的車子給這個乘客，若有多輛車的適配度都同為最高，則指派其中距離最近者，若還有平手則指派編號最小的。我們稱呼 k 為「理想單趟評分」，例如若 $k = 4$ ，則單趟若沒得到 4 分或 5 分，這個司機下次被指派的機會就會變小； h 被稱為標籤單位得分，亦即每個相同的標籤會使此車多加幾分； p 則是距離權重，表現出距離拉遠 1 公里時要扣幾分。

所有上線中的車子與乘客的位置都會被平臺記錄。我們假設這個城市的每個位置都是直角座標系上的格子點，亦即可以用兩個整數 x 和 y 表示。我們也假設這個城市一切的道路都是南北向或東西向的，因此在 (x_1, y_1) 與 (x_2, y_2) 的乘客和車子，他們的距離為 $d = |x_1 - x_2| + |y_1 - y_2|$ 。

一般而言，空車會選擇在路上移動，或者在原地等待。我們假設每一輛空車都會處在五種移動方式的其中一種：向東移動中、向西移動中、向南移動中、向北移動中、原地不動。如果空車在移動中，我們假設普通車總是每分鐘移動恰好一公里，高檔車則是每分鐘恰好移動兩公里。

在本題中，我們將會給你此平臺上發生的一連串事件。每個事件都有一個發生的時間、發生的對象、發生的地點以及事件內容。事件有以下幾類：

- 新乘客註冊：一個新乘客提交他的手機號碼註冊了一個新帳號，選擇數個標籤貼在自己身上，並進入離線狀態。
- 新車子註冊：一個新司機提交他的車牌號碼註冊了一個高檔車或普通車的新帳號，選擇數個標籤貼在自己身上，並進入離線狀態。
- 車子上線：一個已經註冊過的車子在某個地點上線，並進入空車狀態。
- 空車改變移動方式：一輛空車從原本的移動方式改變成新的移動方式，例如左轉、右轉、迴轉、停下、啟動等等。
- 乘客上線：一個已經註冊過的乘客在某個地點上線，並發出對高檔車或普通車的叫車指示。如果 K 公里內沒有空車，他會立刻離線，反之則平臺會指派一輛車給此乘客，該車將開往該乘客處，使乘客的狀態進入等待中，而車子的狀態進入服務中。
- 車子接到乘客：一個被指派給某車子的司機抵達該乘客處，使乘客的狀態進入服務中。
- 車子載著乘客抵達目的地：車子與乘客在某個地點完成服務，乘客進入離線狀態、車子進入空車狀態，平臺則收取此趟服務的費用。車子一定會依照南北向和東西向的規定，走兩點間的最短距離。乘客可能會對車子的這趟服務評分。
- 車子離線：某輛空車決定離線。偶爾我們會遇到非空車說要離線，此時他的離線企圖不會被實現，他的狀態也不會改變。
- 查詢車子狀態：給定某輛車的車牌號碼，印出其狀態（離線、空車、往被配對之乘客前進中、往目的地前進中）、位置（如果為空車狀態）、正被配對之乘客的手機號碼（如果為服務中狀態）、生涯總被評分次數與生涯總得分，或印出此車子不存在之錯誤訊息。
- 查詢乘客狀態：給定某個乘客的手機號碼，印出其狀態（離線、等待中、往目的地移動中）、正被配對之車子的車牌號碼（如果為上線狀態），或印出此乘客不存在之錯誤訊息。
- 查詢平臺收益：請印出平臺累積至今的總收益。
- 系統當機：發生了意外錯誤，全體車子與乘客被強制離線，所有進行中的未完成服務都依然被完成，但系統並不知道，也因此沒收到錢。想當然爾，下一個能發生的事件必然是車子上線或乘客上線。

請你實做一個共乘平臺系統，讀取上述事件後，根據每個事件逐一更新系統狀態，並根據查詢事件印出相關資訊。

輸入輸出格式

系統會提供一共 30 組測試資料，每組測試資料裝在一個檔案裡。在第一行記錄著九個整數，依序是派車最大距離 K 、四個價格參數 a_R 、 b_R 、 a_L 和 b_L 、理想單趟評分 k 、標籤單位得分 h 、距離權重 p ，以及屬性標籤的總數 n 。兩個整數間被一個空白字元隔開。在第二行記錄著 n 個屬性標籤，皆為只含有小寫英文字母與空白字元且字元數不超過 30 的字串。兩個屬性標籤之間被一個分號隔開。若 $n = 0$ ，

則此行空白。已知 $1 \leq K \leq 30$ 、 $10 \leq a_R \leq 100$ 、 $1 \leq b_R \leq 10$ 、 $a_R + 1 \leq a_L \leq 200$ 、 $b_R \leq b_L \leq 20$ 、 $0 \leq k \leq 5$ 、 $0 \leq h \leq 100$ 、 $0 \leq p \leq 100$ ，以及 $0 \leq n \leq 20$ 。舉例來說，若前兩行為

```
10 50 2 80 3 4 10 1 4
love programming;hate programming;love baseball;enjoy chatting
```

表示派車最大距離為 10 公里、普通車之計價公式為搭乘 x 公里需支付 $50 + 2x$ 元、高檔車之計價公式為搭乘 x 公里需支付 $80 + 3x$ 元、理想單趟評分為 4、共有一個標籤加 10 分、距離增加 1 公里扣 1 分，且共有 4 個屬性標籤，分別是「love programming」、「hate programming」、「love baseball」與「enjoy chatting」。

從第三行起，每一行記錄著一個事件，由早排到晚，都在同一天之內。一分鐘內可能會發生很多件事，若是如此則先輸入的事件先發生。每一行的第一個值是一個型態為「hh:mm」的長度為五的字串，以 24 小時制表現幾點幾分，亦即 hh 的值可能是 00、01 到 23，而 mm 的值可能是 00、01 到 59。在第一個值之後緊接著是一個空白字元，然後是第二個值，事件代碼。事件代碼是一個長度為 2 的大寫英文字元。各種事件、其事件代碼與其後輸入的資訊說明如下：

- 新乘客註冊：本事件字串由事件代碼 NP 開始，其後是一個冒號，接著是一個新乘客的手機號碼（長度為 10、只含有數字字元的字串），接著是一個左括號，接著是用逗點隔開的數個屬性標籤，最後是一個右括號。如果此手機號碼尚未註冊過，請在平臺上新增一位乘客用戶，將之設定為離線狀態；反之則跳過此行，不做任何處理。舉例來說，

```
00:01 NP:0987654321(love baseball,enjoy chatting,hate programming)
```

是一個手機號碼 0987654321 的新乘客在 00:01 註冊，他選了三個屬性標籤「love baseball」、「enjoy chatting」和「hate programming」。如果一個乘客沒有選擇屬性標籤，則左右括號中間沒有字元。

- 新車子註冊：本事件字串由事件代碼 NC 開始，其後是一個冒號，接著是一個新車子的車牌號碼（長度為 6、只含有英文或數字字元的字串），接著是一個左括號，接著是用逗點隔開的數個屬性標籤，接著是一個右括號，最後是一個字元 R 或 L 代表普通車或高級車。如果此車牌號碼尚未註冊過，請在平臺上新增一位車子用戶，將之設定為離線狀態；反之則跳過此行，不做任何處理。舉例來說，

```
00:08 NC:AAA111(love baseball)R
00:10 NC:BBB111(hate programming,enjoy chatting)R
```

是一個車牌號碼 AAA111 的新車子在 00:08 註冊，他選了一個屬性標籤「love baseball」，以及一個車牌號碼 BBB111 的新車子在 00:10 註冊，他選了兩個屬性標籤「hate programming」和「enjoy chatting」。如果一輛車子沒有選擇屬性標籤，則左右括號中間沒有字元。

- 車子上線：本事件字串由事件代碼 OC 開始，其後是一個冒號，接著是一個車子的車牌號碼，接著是一個左括號，接著是其上線地點的 x 與 y 座標，中間以逗點隔開，接著是一個右括號，最後是一個英文字元，以 N、S、E、W、H 分別代表往北開、往南開、往東開、往西開與原地停下。如果此車牌號碼已經註冊過且原本不在線上，請在平臺上將此車於該處設定為空車狀態，並讓它向該指定方向行駛；反之則跳過此行，不做任何處理。舉例來說，

```
00:12 OC:AAA111(1,5)N
00:13 OC:BBB111(5,8)N
```

是車牌號碼 AAA111 的車子在 00:12 於 (1,5) 上線，向北行駛，以及車牌號碼 BBB111 的車子在 00:13 於 (5,8) 上線，向北行駛。

- 空車改變移動方式：本事件字串由事件代碼 EC 開始，其後是一個冒號，接著是一個車子的車牌號碼，接著是一個左括號，接著是一個英文字元，以 N、S、E、W、H 分別代表往北開、往南開、往東開、往西開與原地停下，接著是一個右括號。如果此車牌號碼已經註冊過且原本是空車狀態，請在平臺上將此車轉向；反之則跳過此行，不做任何處理。舉例來說，

00:19 EC:AAA111(E)

是車牌號碼 AAA111 的車子在 00:19 轉向東行駛。請注意因為距離此車於 00:12 於 (1,5) 上線後過了 7 分鐘，所以我們知道此車是在 (1,12) 轉向東的。

- 乘客上線：一個已經註冊過的乘客在某個地點上線，並發出對高檔車或普通車的叫車指示。如果 K 公里內沒有空車，他會立刻離線，反之則平臺會指派一輛車給此乘客，該車將開往該乘客處，使乘客的狀態進入等待中，而車子的狀態進入服務中。

本事件字串由事件代碼 OP 開始，其後是一個冒號，接著是一個乘客的手機號碼，接著是一個左括號，接著是其上線地點的 x 與 y 座標，中間以逗點隔開，接著是一個右括號，最後是一個英文字母 R 或 L，代表他想找普通車或高檔車。如果此手機號碼已經註冊過且原本是離線狀態，請在平臺上幫乘客按照規則做搜尋與指派；反之則跳過此行，不做任何處理。舉例來說，

00:23 OP:0987654321(6,10)L

是手機號碼 0987654321 的乘客在 (6,10) 上線搜尋高檔車。由於距離 10 公里內沒有高檔車，此乘客會立刻離線。如果

00:24 OP:0987654321(6,10)R

則表示此乘客又立刻上線搜尋普通車，此時平臺會發現 AAA111 在 (6,12)、BBB111 在 (5,19)，與此乘客的距離都在 10 公里內，便計算這兩輛車與此乘客的適配度。AAA111 的分數為 $0 - 4 \times 0 + 10 \times 1 - 1 \times 2 = 8$ ，而 BBB111 的則為 $0 - 4 \times 0 + 10 \times 2 - 1 \times 10 = 10$ ，所以平臺將指派 BBB111 給乘客 0987654321，BBB111 的狀態變成服務中、0987654321 的則變成等待中。

- 車子接到乘客：本事件字串由事件代碼 CP 開始，其後是一個冒號，接著是一個車子的車牌號碼。如果此車牌號碼已經註冊過、原本是服務中狀態、與其配對的乘客是等待狀態，且事件時間晚於該車子與該乘客配對的時間，請在平臺上將此乘客設定為服務中狀態；反之則跳過此行，不做任何處理。

當車子確實接到乘客時（亦即這一行的事件要被處理時），請進一步檢查乘客的等待時間。如果乘客叫了車，結果叫車到被接到的時間大於 10 分鐘且小於等於 20 分鐘，本趟車資照算但乘客將會給予車子較差的評價；如果此時間大於 20 分鐘，平臺將不向乘客收取本趟車資。

舉例來說，

00:30 CP:BBB111

是車牌號碼 BBB111 的車子在 00:30 接到他被指派的乘客，因此 0987654321 的狀態要變成服務中。乘客等待了 6 分鐘，所以車資照算，乘客也對等待時間感到滿意³。

³你或許覺得奇怪，十公里的路程，怎麼忽然只要 6 分鐘就到了？你可以想像，司機平時空車時在街上慢慢晃，有生意做了就開快一點，也是很合理的。總之，只要接客人的時間晚於客人叫車的時間，我們就一律當成合理。

- 車子載著乘客抵達目的地：本事件字串由事件代碼 **AD** 開始，其後是一個冒號，接著是一個車子的車牌號碼，接著是一個左括號，接著是其抵達地點的 x 與 y 座標，中間以逗點隔開，接著是一個右括號，最後是一個英文字元，以 **N**、**S**、**E**、**W**、**H** 分別代表此空車將往北開、往南開、往東開、往西開與原地停下。如果此車牌號碼已經註冊過、原本是服務中狀態、與其配對的乘客是服務中狀態，且此事件的時間晚於車子接到乘客的時間，則請在平臺上將此乘客設定為離線、車子設定為空車且向指定方向行駛，為平臺累積收益，且讓乘客幫車子評分；反之則跳過此行，不做任何處理。

平臺能累積的收益以及乘客幫車子的評分，取決於乘客的等待時間、從起點到目的地的移動時間，以及司機和乘客相同的屬性標籤個數。關於收益，首先如前所述，如果乘客的等待時間太長，則本趟不收車資。與之類似地，假設乘客上車的地點與目的地間的距離為 d ，則預期移動時間是 $\mu = d$ （普通車）或 $\lceil \frac{d}{2} \rceil$ （高檔車）。如果確實的移動時間大於 3μ ，則本趟也不收車資。換言之，只要乘客的等待時間超過 20 分鐘，或搭乘時間超出過預期時間的三倍，這趟旅程就免費。

關於評分，我們假設乘客都會用以下方式幫車子評分。如果沒有特別原因，乘客會給車子的基本分是 4 分，但如果他的等待時間大於 10 分鐘且小於等於 20 分鐘則會扣 1 分，大於 20 分鐘則會扣 2 分；如果車子運送她的時間大於 2μ 且小於等於 3μ 則會扣 1 分，大於 3μ 則會扣 2 分。最後，車子和乘客每有一個相同的屬性標籤，就加 1 分。這樣加加減減後的分數若超過 5 分則以 5 分計，低於 1 分則以 1 分計。

舉例來說，

00:42 AD:BBB111(6,14)S

是車牌號碼 BBB111 的車子在 00:42 將 0987654321 送抵 (6,14)，並開始面向南方準備開車。剛剛已經計算過，乘客的等待時間 $6 \leq 10$ ，這方面不扣分。因為移動距離是 4 公里，對此普通車的預期運送時間為 $\mu = 4$ ，而實際運送時間是 $12 \leq 3\mu$ ，因此會扣 1 分。司機和乘客有 2 個屬性標籤相同，因此加 2 分。綜合以上，乘客對車子的評分為 5 分。本趟服務平臺共收益 $50 + 2 \times 4 = 58$ 元。

當然，因為一分鐘內可能會發生好幾個事件，所以如果 AD 事件早於同一分鐘發生的 ZZ 事件，那那筆收益就要被累積；如果 AD 晚於 ZZ，就不被累積。

- 車子離線：某輛空車決定離線。偶爾我們會遇到非空車說要離線，此時他的離線企圖不會被實現，他的狀態也不會改變。本事件字串由事件代碼 **LC** 開始，其後是一個冒號，接著是一個車子的車牌號碼。如果此車牌號碼已經註冊過且原本是空車狀態，則將此車設為離線；反之則跳過此行，不做任何處理。舉例來說，

00:45 LC:AAA111

表示 AAA111 在 00:45 離線。

- 查詢車子狀態：本事件字串由事件代碼 **SC** 開始，其後是一個冒號，接著是一個車子的車牌號碼。如果此車牌號碼已經註冊過，則印出該車牌號碼，接著一個空白字元，接著一個整數代表此車的狀態（離線為 0、空車為 1、往被配對之乘客前進中為 2、往目的地前進中為 3），接著一個空白字元，接著一個整數代表生涯總被評分次數，接著一個空白字元，接著一個整數代表生涯總得分。如果是空車狀態，則接著依序印出一個空白字元、車子位置的 x 座標、一個空白字元、車子位置的 y 座標；如果是服務中狀態，則接著依序印出一個空白字元與被配對之乘客的手機號

碼。如果此車牌號碼沒有註冊過，則印出該車牌號碼，接著一個冒號，接著一個空白字元，接著「no registration!」。不論車子是否有註冊，都要在印出的資訊最後印一個換行符號。舉例來說，

```
00:49 SC:AAA111
00:50 SC:BBB111
00:51 SC:CCC111
```

應該印出

```
AAA111 0 0 0
BBB111 1 1 5 6 6
CCC111: no registration!
```

表示 BBB111 當下為空車、生涯共計被評分一次、生涯累計總分為 5 分、當下位置在 (6,6)。

- 查詢乘客狀態：本事件字串由事件代碼 SP 開始，其後是一個冒號，接著是一個乘客的手機號碼。如果此手機號碼已經註冊過，則印出該手機號碼，接著一個空白字元，接著一個整數代表此乘客的狀態（離線為 0、等待中為 1、往目的地前進中為 2）。如果是上線狀態，則接著依序印出一個空白字元與被配對之車輛的車牌號碼。如果此手機號碼沒有註冊過，則印出該手機號碼，接著一個冒號，接著一個空白字元，接著「no registration!」。不論此乘客是否有註冊，都要在印出的資訊最後印一個換行符號。舉例來說，

```
00:52 SP:0988123456
```

應該印出

```
0988123456: no registration!
```

- 查詢平臺收益：本事件字串由事件代碼 SR 開始，沒有別的字元。此時請印出平臺累積至今的總收益，最後印一個換行符號。舉例來說，

```
00:57 SR
```

應該印出

```
58
```

- 系統當機：發生了意外錯誤，全體車子與乘客被強制離線，所有進行中的未完成服務都依然被完成，但系統並不知道，也因此沒收到錢。

本事件字串由事件代碼 ZZ 開始，沒有其他字元。發生此事件的當下，所有已註冊的車輛或乘客的狀態都要立刻變成離線。舉例來說，

```
00:55 ZZ
00:56 SC:BBB111
```

應該印出

```
BBB111 0 1 5
```

讀入資料後，請一行一行地處理發生的事件，並且在需要時印出結果。舉例來說，如果輸入是

```
10 50 2 80 3 4 10 1 4
love programming;hate programming;love baseball;enjoy chatting
00:01 NP:0987654321(love baseball,enjoy chatting,hate programming)
00:08 SC:AAA111
00:08 NC:LLL111()L
00:08 NC:AAA111(love baseball)R
00:10 NC:BBB111(hate programming,enjoy chatting)R
00:12 OC:AAA111(1,5)N
00:13 OC:BBB111(5,8)N
00:13 OC:LLL111(6,10)H
00:19 EC:AAA111(E)
00:23 LC:LLL111
00:23 OP:0987654321(6,10)L
00:24 OP:0987654321(6,10)R
00:30 CP:BBB111
00:42 SR
00:42 AD:BBB111(6,14)S
00:42 SR
00:45 LC:AAA111
00:49 SC:AAA111
00:50 SC:BBB111
00:51 SC:CCC111
00:52 SP:0988123456
00:53 NP:0999888777()
00:53 OC:LLL111(20,20)H
00:53 OP:0999888777(20,20)L
00:53 CP:LLL111
00:55 ZZ
00:56 SC:BBB111
00:57 OC:AAA111(9,8)S
00:57 SR
```

請注意 00:08 共有三個事件，依序是查詢 AAA111、LLL111 註冊與 AAA111 註冊，因為查詢 AAA111 先發生，當時 AAA111 還未註冊（即使在同一分鐘，事件依然有先後之別），所以此時要印出 AAA111: no registration!。也請注意 00:23 共有兩個事件，LLL111 先離線，接著 0987654321 搜尋高檔車，但因為 LLL111 已經離線了，因此 0987654321 沒有得到高檔車配對。00:42 也有三個事件，其中第一個事件在查詢平臺收益，此時收益為零；在第二個事件成功地將乘客載到目的地、平臺收入 58 元後，在第三個事件再次查詢平臺收益就要印出 58 最後，00:55 的系統當機讓 00:53 開始的乘客運送無法在平臺上被完成，因此不會增加收入。

綜合以上，我們應該印出


```
AAA111: no registration!
0
58
AAA111 0 0 0
BBB111 1 1 5 6 6
CCC111: no registration!
0988123456: no registration!
BBB111 0 1 5
58
```

本題中事件的個數不限，每筆測試資料中註冊的車輛或乘客數也不限。你可以假設所有需要被記錄的數字，都可以被存在 `int` 而不會溢位。

評分原則

這一題的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。測試資料的複雜度資訊如下：

- 前 10 筆測試資料中所有空車都只會待在原地不動、屬性標籤個數為零，且系統不會當機。
- 第 11 到 20 筆測試資料中所有空車都只會待在原地不動，且系統不會當機。
- 第 21 到 25 筆測試資料中系統不會當機。
- 第 26 到 30 筆測試資料中所有事情都可能發生。

第二題

(20 分) 請為你在第一題設計的各個 class 寫一份給開發者的說明文件，讓其他開發者（或未來的你）能快速理解你設計的這些 class。你只需要說明你的設計，不需要說明你的實做，因此你應該：

1. 展示你的那些 class，並且說明繼承與多型的關係（如果有的話）。
2. 對每個 class 列出其所有成員（instance variable、static variable、instance function、static function）的所有資訊（變數的類別、名稱、初始值等；函數的完整 header）。
3. 針對每一個成員變數，說明其在該 class 中作何用途。
4. 針對每一個成員函數，說明其參數的性質與用途、其回傳值的性質與用途，以及呼叫此函數前後你的成員變數會有何變化。
5. 這些 class 中若有用到樣板和例外處理，也請說明之。
6. 加上其他應該補充的資訊。

你的說明應該精確而簡潔（precise and concise）。所謂的精確就是不模糊（廢話?!），例如「執行完此函數後，某某陣列內的整數會被排序」就比「執行完此函數後，某某陣列內的整數會被由小到大排序」不精確。所謂的簡潔就是用字精要、少說廢話。這份文件**不能超過四頁**，超過的部份不列入批改。

第三題

(20 分) 第一題截止後，我們會讓同學們互相檢視彼此的本題程式碼。請在 PDOGS 上批改你被隨機分配到的第一題程式碼，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 main function 是否能很快地理解程式在大方向上的運算邏輯？
- class 設計：是否有宣告與設計合適的 class？這些 class 是否都確實應該被用來生成物件？是否提供了太多不必要的 public function 因而破壞了 data hiding？是否把做為 global function 更合適的函數寫成了 member function，或把做為 member function 更合適的函數寫成了 global function？
- 進階程式設計方法：你應該檢查那份程式碼有沒有合理地使用封裝、繼承、多型、樣板和例外處理，如果沒有就扣他一分。當然，請明確地指出他應該錯過的使用時機。
- 資料結構技巧：你應該檢查那份程式碼有沒有合理地使用 bag 資料結構，如果沒有就扣他一分。當然，請明確地指出他應該錯過的使用時機。
- 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。

本題其中 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）；若你在本次作業中完全沒有寫第一題，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。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；只要你有完成評語和評分，且沒人檢舉你或有人檢舉你但助教檢視後認為你沒問題，你就會得到這 10 分。