

程式設計 (106-1)

作業九

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、二題上傳一個 PDF 檔，再為第三、四題各上傳一份 C++ 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 11 月 28 日凌晨一點**。在你開始前，請閱讀課本的第 7.1-7.9 和 7.11 節¹。為這份作業設計測試資料並且提供解答的助教是林翰伸。

第一題

(20 分)

(a) (5 分) 請考慮下面這個拙劣的程式：

```
#include <iostream>
using namespace std;

int main()
{
    int a = 10;
    int* ptr = &a;
    cout << ptr[1];
    return 0;
}
```

請解釋為什麼 `ptr` 明明不是一個陣列，也不是一個指向動態陣列的指標，編譯器卻能理解並且接受 `ptr[1]` 這樣的語法，以及 `ptr[1]` 的意涵。

(b) (5 分) 承上題。若我們執行該程式，在 `cout << ptr[1];` 有可能會遇到 run-time error 嗎？為什麼？

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

(c) (10 分) 請考慮下面這個更拙劣的程式：

```
#include <iostream>
using namespace std;

void f(int* ptr);
void g(int*& ptr);

int main()
{
    int a = 10;
    int* ptr = &a;
    f(ptr);
    cout << ptr[1];
    g(ptr);
    cout << ptr[0];
    return 0;
}

void f(int* ptr)
{
    ptr = nullptr;
}

void g(int*& ptr)
{
    ptr = nullptr;
}
```

為什麼執行完 `f(ptr)`；並且在函數裡把 `ptr` 設成 `nullptr` 後，執行 `cout << ptr[1]`；不是一定會發生 run-time error？又為什麼執行完 `g(ptr)`；並且在函數裡把 `ptr` 設成 `nullptr` 後，執行 `cout << ptr[0]`；就一定會發生 run-time error？

提示：call by value 和 call by reference！

第二題

(20 分) 請在 PDOGS 上批改你被隨機分配到的作業六第三題的程式碼，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、structure 或 class²？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？
- 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平，否則只要有評就會得到 10 分）。

第三題

（60 分）很多城市都有有樁的公共自行車（或稱共享自行車），在臺北就是 YouBike。假設在一個城市裡有 N 個公共自行車站，第 i 個站共有 n_i 個車架，若我們隨時去觀察第 i 個站正有 x_i 輛車停在那裡，則必有 $0 \leq x_i \leq n_i$ 。假設此刻系統中一共有 m 輛車。由於每一輛車都有各自的狀態（例如是否被租借中、是否故障等），因此系統必須給每一輛車一個唯一的編號，介於 1 到 m 之間。才能隨時記錄每一輛車的此刻位置（在某個站上或被租借中），對被租借中的車才能記錄它是從哪個站被借的。在本題中，我們將實做一個簡單的公共自行車管理系統，在給定車輛初始配置狀態以及一系列借還交易記錄後，能隨時得知各站的車數與各車的狀態。

題目會先給定車站數 N 、總車輛數 m 、車站 i 的車架數 n_i ，以及在初始時刻下每一輛車的狀態。自行車 j 的狀態包含現在狀態 y_j 與所屬車站 z_j ，其中 y_j 若為 1 表示這輛車正被租借中，為 0 則表示正停在車架上。若 $y_j = 0$ ， z_j 自然就表示這輛車停在哪個車站；若 $y_j = 1$ ，則 z_j 表示這輛車是從哪個站被借走的。實務上我們應該是記錄這輛車停在哪個車站的哪個車架上。為了簡單起見，在本題我們忽略車架資訊。

題目接著會給定一系列共 K 個借還交易記錄。在第 k 筆交易記錄中，我們會記錄車輛編號 u_k 與交易狀態 t_k （0 表示借、1 以上表示還車的車站編號）。實務上我們還會記錄借還車的會員卡號，以及借還時間。在本題中我們忽略這兩個資訊。

²我們還沒教 structure 和 class，所以本次批改請看函數就好。

在給定初始狀態與一系列的借還記錄後，請由小到大印出每一個車站中正停著的所有車輛的編號。

說明：由於各站的車架數不同，使用指標來實做各列長度不同的二維動態陣列，是很合理的作法。請在此題練習這種作法，並且注意本題最後面的相關說明。

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。在每個檔案中會有若干行，第一行包含三個整數 N 、 m 和 K ，第二行包含 N 個整數 n_1 、 n_2 直到 n_N ，第三行包含 m 個整數 y_1 、 y_2 直到 y_m ，第四行包含 m 個整數 z_1 、 z_2 直到 z_m 。從第五行起共有 K 行，第 $k+4$ 行包含兩個整數 u_k 和 t_k 。同一行的任意兩個值之間被一個空白隔開。已知 $1 \leq N \leq 100$ 、 $1 \leq m \leq 10000$ 、 $0 \leq K \leq 10000$ 、 $1 \leq n_i \leq 50$ 、 $y_j \in \{0, 1\}$ 、 $1 \leq z_j \leq N$ 、 $1 \leq u_k \leq m$ 、 $t_k \in \{0, 1\}$ 。借還交易不會出現任何不合理的事情，像是一輛車還沒被借就被還、車架已滿但依然有人還車等等。

讀入資料後，請根據給定資訊計算車輛的借還與移動，並求得最終的系統狀態。接著請印出 N 列資訊，第 i 列先印出正停在車站 i 的總車數，接著印出一個冒號（冒號和總車數之間沒有空白），接著是由小到大排列的正停在車站 i 的所有車輛的編號。任兩個編號之間用一個空白字元隔開。最後一個編號後面不應該有空白，但應緊接著一個換行字元。如果該車站沒有任何車輛正停在那邊，該行的冒號後面就只有一個換行字元。舉例來說，如果輸入是

```
3 5 4
10 10 5
1 0 0 0 0
2 1 2 2 1
3 0
4 0
3 2
1 1
```

則輸出應該是

```
3:1 2 5
1:3
0:
```

請注意第三個車站沒有任何車輛停靠，因此就印出 0:。如果輸入是

```
3 5 0
10 10 5
1 0 0 0 0
2 1 2 1 3
```

則輸出應該是

```
2:2 4
1:3
1:5
```

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。

除此之外，你**不可以**只使用靜態陣列去完成此題。你**必須**使用動態陣列去儲存站點內停靠之車輛資訊。若你沒有使用指標與動態陣列，「程式的品質」部份將被扣分。

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。
- 這一題的其中 20 分會在作業十中被評定。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，現在給定的 K 個交易記錄中包含著三類交易：合理的交易記錄、不合理的交易記錄、擴充車架數的記錄。合理的交易記錄就是第三題那些；不合理的交易包含以下幾種：

- 企圖借某輛正在被租借中的車。
- 企圖還某輛沒有被租借中的車。
- 企圖把某輛被租借中的車還到某個已經滿了的车站。

最後，車站的車架數 n_i 可能會被增加。請依然處理這 K 個交易記錄，並在最後按照第三題的規定印出最終的系統狀態。若某個交易記錄不合理，就忽略這一個交易記錄。

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。在每個檔案中會有若干行，其中前四行的格式和第三題一樣。從第五行起共有 K 行，第 $k+4$ 行包含一個大寫英文字元和兩個整數。若該英文字元為 B，則兩個整數依第三題格式表示車輛編號和交易記錄；若該英文字元為 S，則第一個整數 a_k 是一個車站的編號，第二個整數 b_k 則表示此車站的車架數的增加數量。同一行的任意兩個值之間被一個空白隔開。已知 $a_k \in \{1, \dots, N\}$ 、 $1 \leq b_k \leq 10$ 。

舉例來說，如果輸入是

```
3 7 4
10 10 5
1 0 0 0 0 0 1
2 3 3 3 3 3 1
B 1 3
```

```
B 7 1
S 3 5
B 1 3
```

則輸出應該是

```
1:7
0:
6:1 2 3 4 5 6
```

請注意第一個交易應該被忽略，因為當時車站 3 已經沒有空位了，但一模一樣的交易出現在第四個交易時，就會被執行，因為第三個交易在第三個車站擴充了五個車架。

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。