

# 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/>) 為每一題各上傳一份 C++ 原始碼（以複製貼上原始碼的方式上傳）。第三題是 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 **11 月 20 日凌晨一點**。在你開始前，請閱讀課本的第 7.10 和 22.9–22.13 節<sup>1</sup>。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）針對以下十題是非題，我們會使用 PDOGS 自動批改，因此請寫一個 C++ 程式，內容就是先讀入一個整數，若讀入的數字為  $i$ ，則印出第  $i$  小題的答案，若為是則印出 1、若為否則印出 0。舉例來說，如果題目只有四題，且你認為答案依序是是、否、是、是，則你上傳的程式碼應該是

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

int main()
{
    int problem = 0;
    cin >> problem;
    if(problem == 1)
        cout << 1;
    else if(problem == 2)
        cout << 0;
    else if(problem == 3)
        cout << 1;
    else
        cout << 1;

    return 0;
}
```

PDOGS 會餵給你的程式的，一定是 1、2 直到 10 這十個整數。有別於作業中一般的程式題，本題在你上傳程式碼時，測試資料是還沒有放上 PDOGS 的，助教會等作業截止後才上傳測試資料（和答案）到 PDOGS 並重新批改此題。換言之，你上傳程式碼時是不會顯示你得幾分的，更不會顯示你對或錯哪些筆測試資料。你會看到你得 0 分，但此數字在助教重新批改之後就會被更新成正確的分數了。

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

以下題目如果沒有特別指名，請用 C++ 為基準作答。若你看到一段程式碼，請假設他們是被寫在一個有良好且必備的 include 敘述、using namespace 敘述的程式的結構正確的 main function 裡面。

- (a) `isalpha` 這個函數，雖然是回傳 `int`，但其實用 `bool` 變數去接回傳值，是完全沒有問題的。
- (b) C++ 中的字元陣列，只要在最後面有 `\0`，就可以表示一個字串。
- (c) C++ 中用來標示字元和字串的引號，單引號跟雙引號都個別可以用來標示字元和字串，只要不要單引號配雙引號就好。
- (d) 用 `cin >>` 去寫入一個字串時，系統會自動在最後面加入一個 `\0`，而寫入一個字元時則不會。
- (e) 用 `cin.getline()` 讀入字串時，預設是不會用空白字元切割字串，用 `cin >>` 則會。
- (f) 不考慮電腦壞掉、記憶體不夠之類的問題，執行

```
char* p = "12345";  
cout << p + 2 << "\n";
```

一定會看到 345，沒有例外。

- (g) 不考慮電腦壞掉、記憶體不夠之類的問題，執行

```
char* p;  
p = "12345";  
cout << p + 2 << "\n";
```

一定會看到 345，沒有例外。

- (h) 不考慮電腦壞掉、記憶體不夠之類的問題，執行

```
char p[100];  
p = "12345";  
cout << p + 2 << "\n";
```

一定會看到 345，沒有例外。

- (i) 不考慮電腦壞掉、記憶體不夠之類的問題，執行

```
char p[100];  
strcpy(p, "12345");  
cout << p + 2 << "\n";
```

一定會看到 345，沒有例外。

- (j) 下面的函數

```
void swapPtr(char*& p1, char*& p2)  
{  
    char* temp = p1;  
    p1 = p2;  
    p2 = temp;  
}
```

因為 `*` 和 `&` 是互為反運算，所以把 header 改成

```
void swapPtr(char p1, char p2)
```

結果也會完全不變。

**小提醒：**在 PDOGS 上面讓大家繳交此題的地方，會有兩組「與上面正式要計分的題目完全無關的」範例輸入輸出，純粹是用來讓大家確認自己那個被批改的 `if-else` 程式是可以被正確執行的。請確認你的程式在針對範例輸入輸出做撰寫後，能讓你在這一題得到「Accepted」，接著再去針對要計分的題目把你的正確答案寫上去然後繳交。當然，即使你曾經看到「Accepted」，也不代表你繳交的題目在這題已經得到滿分了。

## 第二題

(60 分) 這年頭，零售商當然都有使用 POS (point of sales) 銷售系統，去記錄每一筆銷貨記錄。通常一筆銷貨記錄會有以下三個最主要的元素：被賣出的產品的品名、賣出的時間 (或日期)，以及賣出的數量。有了這長時間累積下來的許多筆資料，就可以做一些分析 (例如銷售預測)。

話說如此，很多時候資料的品質會是一個問題，而一個經常在零售商發生的資料品質問題，是品名的混亂。舉例來說，授課教師本人曾經做過一個零售商的委託研究案，他們的產品品名就有「iPhone5」、「iPhone 5」、「iphone-5」、「iphone5-32G」、「iphone-5 32G」這些，族繁不及備載。產生這些品名混亂的原因很多，這裡也不討論，我們就想做一件事：做任何分析前，要把其實是同一個產品只是資料庫中品名不同的記錄合併起來。

在本題中，為了簡化題目，我們考慮授課教師本人曾經做過的另一個委託研究案中的問題。這個公司的產品如果名稱是 XXX，資料庫中會有 XXX、XXX-vip (賣給 vip 的 XXX)、XXX-special (從特別管道取得的 XXX)、XXX-old (舊物新用的 XXX) 等，但其實在倉庫裡都算是同一種東西。因此，如果在某個時段內，這家公司賣掉了 4 個 XXX、3 個 XXX-vip、2 個 XXX-special 和 1 個 XXX-old，其實就是賣掉了 10 個 XXX 的意思。此外，雖然產品名稱都是由英文字母、數字和連字號組成，在英文字母方面也會有大寫誤打成小寫、小寫誤打成大寫的問題。資料庫中只有大小寫差別的不同產品名稱，也應該對應到同一種商品。例如 XXX 和 xxx 和 XXx 和 xxx-old 和 XxX-Old 都是同一種商品。

針對給定的許多交易記錄，請考慮上述情況將產品名稱做彙總，接著就給定的起訖日期，計算所有彙總後產品的該期間內銷售量 (包含那兩天)，並將這段期間內有銷售至少一個的所有產品及其銷售量，依照產品名稱的字典順序 (就是 `strcmp` 會回傳的順序) 由先而後印出。

## 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。每個檔案中有  $n + 2$  行，其中第一行是一個正整數  $n$ ，第二行是兩個以 `yyyymmdd` 呈現的日期字串  $d_0$  和  $d_1$ ，中間被一個空白字元隔開。第三行起的  $n$  行則每行代表一個交易記錄，依序含有一個字串  $x$  代表產品名稱、一個逗點、接著一個以 `yyyymmdd` 格式呈現的日期字串  $d$ 、一個逗點、一個整數  $s$  代表該產品該日銷售量，最後一個換行符號。 $x$  中只含有最多一個連字號「-」，其餘字元都是大寫或小寫英文字元或數字。已知  $n \in \{1, \dots, 1000\}$ 、 $x$  的長度介於 1 到 50 之間、 $d$ 、 $d_0$  和  $d_1$  都是合法的日期字串且介於 2010/1/1 和 2020/12/31 之間，以及  $s \in \{1, \dots, 100\}$ 。任兩筆交易記錄若不是產品名稱不同，就是日期不同。

讀入這些資料後，請依照題意做資料彙總，去計算介於  $d_0$  和  $d_1$ （包含這兩天）之間的所有產品的總銷售量，並且把 XXX、XXX-vip、XXX-old、XXX-special 這些產品名稱都視為同一種產品 XXX（XXX 可以被任何只含有英數字元的字串代入）。只有大小寫不同的產品名稱視為同一種（例如 Apple 和 appLE 是同一種產品，apple-VIP 和 apple-vip 也是同一種產品，以上四個都是同一種產品）。彙總完成後，請將所有產品依照字典順序由先到後排序，並每行一個印出每個產品的產品名稱（不含連字號以及後面的綴字）及其搜尋日期區間內的總銷售量。產品名稱的英文字元都以小寫印出。產品名稱與總銷售量之間用一個冒號隔開，總銷售量後面加上一個換行符號。

舉例來說，如果輸入是

```
10
20180101 20181010
APPLE,20180101,1
orange,20180301,4
orange-vip,20180302,8
orange-special,20180303,5
orange-vip,20190301,400
banana-old,20180601,1
apple,20180601,1
bananananana,20200601,1
baNana-delicious,20180601,1
banama,20180901,1
```

則輸出應該是

```
apple:2
banama:1
banana:1
banana-delicious:1
orange:17
```

請注意 **banana-delicious** 並不是 **banana**，因為只有連字號後面是 vip、old、special（大小寫不計）的才應該跟沒有連字號的視為同種。此外，2019/3/1 和 2020/6/1 的交易不在搜尋範圍內，所以 **orange** 在 2019/3/1 的 400 個銷量不該被計入，**bananananana** 甚至不出現在輸出清單中。

## 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。

## 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 分會在後續作業中被評定。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。

以下我們列出屆時的同儕互評標準，供你現在撰寫程式參考：

請在 PDOGS 上批改你被隨機分配到的程式碼，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？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。

### 第三題（bonus）

（20 分）承上題，我們要處理一模一樣的問題，但是現在產品名稱中可能有超過一個連字號。而連字號方面則只根據最後一個連字號後面的字串做產品名稱合併的考慮，換言之，最後一個連字號之前的所有字元都被當作是產品名稱的一部分，不被納入產品名稱合併的考量。只有英文字母大小寫不同的產品名稱依然可以被視為同種產品。舉例來說，apple-good、apple-good-vip 是同一種產品，而且他們跟

apple 不是同一種產品；banana-vip-great 跟 banana-vip-great-old 是同一種產品，而且他們跟 banana 或 banana-great 都不是同一種產品。

本題的輸入輸出格式和第三題幾乎一模一樣，唯一的差別只在於產品名稱中可能有超過一個連字號。舉例來說，如果輸入是

```
10
20180101 20181010
APPLE,20180101,1
orange,20180301,4
orange-vip,20180302,8
orange-special,20180303,5
orange-vip,20190301,400
banana-old,20180601,1
apple,20180601,1
banana-vip-great,20180601,1
BANANA-vip-great-OLD,20180601,1
banana-great,20180901,1
```

則輸出應該是

```
apple:2
banana:1
banana-great:1
banana-vip-great:2
orange:17
```

針對這個題目，你可以使用任何方法。這一題的 20 分會根據程式運算的正確性給分，一筆測試資料佔 2 分。