

程式設計 (105-2)

作業七

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，再為第三題至第五題各上傳一份 C++ 原始碼 (以複製貼上原始碼的方式上傳)。第四題跟第五題都是 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 4 月 17 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¹。為這份作業設計測試資料並且提供解答的助教是楊佩蓉 (Sophie Yang)。

第一題

(20 分；每小題 10 分) 請回答下列兩題：

- (a) 在投影片第 27 頁，我們有一個二維靜態陣列 `name`。如果我在那五行程式碼後面接著寫

```
char* p = name[0];
cout << p << "\n";
p = name[1];
cout << p + 1 << "\n";
```

請問會看到什麼輸出？為什麼？

- (b) 請用自己的話，解釋投影片第 32 頁的 `reverse()` 函數，並說明為何該頁程式的輸出結果會是 15432。

第二題

(20 分；每小題 10 分) 請回答下列兩題：

- (a) 請自行上網查詢 `strcspn()` 這個函數，用自己的話說明這個函數的使用方式，並且設計一個 C++ 的範例程式去做介紹。

- (b) 請寫一個函數

```
void strFormatting(char* p);
```

去使得傳入的字串 `p` 中的每個 English word 的字首大寫、其他字母小寫。你可以假設 `p` 裡面只含有英文字母 (大小寫共 52 個) 與空白鍵，而所謂的 English word 就是被空白鍵隔開的不含空白鍵的字串。

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

第三題

(60 分) 有一個老闆在一個小鎮經營水果攤，請了許多店員負責結帳和負責記帳。當年還沒數位化的時候，他們就是在一張表格上一筆筆地記錄「蘋果，100」、「橘子，150」，其他還有一些欄位記錄像是時間、結帳員等等的資訊，因為本題用不到，所以就不提了，總之（對我們來說），就是一張表格兩欄，第一欄是商品名稱，第二欄是該筆交易該品項的營業額（賣了多少錢）。值得注意的是，因為老闆是個爽快的人，所以他們也允許消費者退貨（至於怎麼確認該水果真的是從這個攤子買的，就不勞我們費心了），此時就會記錄「香蕉，-80」，也就是用負值來表現退貨時還給消費者的金額。如此一來，到了月底時就只要對每個品項個別地做正負相加，就知道該品項的淨營業額是多少了。

經營多年之後，老闆忽然覺得應該要來對過去幾年的歷史銷售資料做點分析，例如看看各種水果的銷售季節性、銷售相關性等等，此時才發現用紙筆記錄的帳目資料實在難以分析。因此，老闆決定請一位店員幫他做「資料數位化」……其實就是一筆一筆打進一個純文字檔而已²。店員甲充實地把帳本上的所有資訊都輸進電腦之後，店員乙仔細地檢查了，確實是分毫不差。

接著老闆就想請人來做資料分析，此時才發現有個資料格式的問題，就是每個店員記帳時寫數字的方式不一樣：

- 對於超過一千的數字，例如 1234，有的人會寫 1234 但也有的人會寫 1,234。雖然只是個小攤子，偶爾卻會一口氣賣掉價值一百萬元以上的水果，此時第一種人不會加任何逗點，第二種人則會正確地加上兩個逗點。要加逗點的人就會把正確數量的逗點加在正確的地方，不會寫出 12,34、1,2,34、1,,234 或 1234,567 這種東西。
- 有的人會在開頭加上錢字號，像是 \$1234，有的人不會。要加錢字號的人就會加恰好一個錢字號在數字前面，不會加多個錢字號，也不會把錢字號寫在其他地方，所以不會寫出 \$\$1234 或 12\$34 這種東西。
- 因為這個國家的貨幣有元、角、分等等，最小的幣值是 0.01 元（一分），所以有些營業額或退款額是有小數的，最多小數兩位。有些店員在記帳時一律寫滿兩位小數，像是 1234.50 或 1234.00，有些則恰好寫到需要的位數，像是 1234.5 或 1234。
- 負數花樣就更多了，有的人寫負號，像是 -1234，有的人則是用括弧刮起來，像是 (1234)。如果寫負號，就一定是一個負號，不會有多個負號，負號也一定出現在數字或錢字號的左邊。如果有左括弧，就一定有右括弧。括弧只會出現最多一對，不會出現複數對。如果同時有錢字號跟負號，先寫負號；如果同時有錢字號跟左括弧，先寫左括弧。負號和括弧不會同時出現。
- 絕對值大於等於 1 的數字開頭不會是零，絕對值小於 1 的數字會有整數位，其值為零，所以不會出現 012 或 .12 這種東西。
- 整個數字字串裡面不會含有空格，也不會含有上面沒提到的字元。

綜合以上，正數有八種寫法，負數則有十六種寫法。

想當然爾，要做任何分析之前，必須先把資料格式統一，而這就是你的任務了。你將會被給定許多依照上述方式記錄下來的「數字字串」，請寫一個程式將他們全部轉成正確的數字，然後輸出這些數字的最大值、最小值和總和。

²請假設那個世界上沒有 Excel 之類的試算表軟體。

輸入輸出格式

系統會提供許多筆測試資料，每筆測試資料裝在一個檔案裡。在每個檔案中，第一列存放兩個整數 n 和 m ，分別是水果的種類數以及交易的筆數。第二列到第 $n + 1$ 列各存放一個英文字串，只用二十六個小寫英文字母組成。這 n 個字串是 n 種水果的名稱，彼此不重複。第 $n + 2$ 列到第 $n + m + 1$ 列各存放一個字串，依序有三個部份：一個英文字串代表被賣出或退回的水果、一個空白鍵，以及一個代表營業額或退款額的「數字字串」。在這些交易記錄中的英文字串可能包含大寫英文字母，不過如果全部轉成小寫，就一定可以恰好對應到 n 種水果的其中一種。請注意「數字字串」中也可能包含逗點、小數點、負號、錢字號和左右括弧，但無論如何該數字字串一定符合題目敘述中的某一種記錄方式。已知 $1 \leq n \leq 10$ 以及 $1 \leq m \leq 1000$ 。此外，一個水果的名稱長度不超過 20 個字元，而一個數字字串被正確地轉成數字之後，一定落在 -99999999 和 99999999 之間

請讀入這一批資料，並且檢視所有代表營業額或退款額的數字，依序印出這批數字的最大值、最小值和總和，兩個數字之間各以一個空白鍵隔開。不論這三個數字各是整數還是小數，一律印出兩位小數。請以不含負號和小數點以外符號的純數字方式印出。

舉例來說，如果輸入是

```
2 6
apple
banana
Apple $10.0
Banana 10
apple 12.1
banana -$2.1
apple 2,000.00
banana ($1,000.50)
```

則六個交易額或退款額依序是 10、10、12.1、-2.1、2000 和 -1000.5，因此最大值是 2000、最小值是 -1000.5、總和是 1029.5，輸出應該是

```
2000.00 -1000.50 1029.50
```

你不需要檢查數字的合理性（例如退款總額是否高於銷售總額）。

小提醒：如果你想要使用 `atof()` 這個函數，記得確認它的回傳型態是什麼！

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，但有個例外：所有定義在 `<cstring>` 裡面的函數都可以用，即使上課沒有教過。

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。前 30 分由 15 筆測試資料判定分數，一筆測試資料佔 2 分；後

10 分由 5「組」測試資料判定分數，每一組裡面有若干筆測試資料，全對的話才能得到 2 分。

- 這一題的其中 20 分會在作業八中被評定。在作業八中，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）。該任務在作業七中會佔 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。若你在作業七中沒有寫這一題，那作業八自然沒有人能檢視你的程式碼，你也就得要損失那 10 分了。

第四題 (bonus)

(20 分) 承第三題，我們現在要對各個水果分別做統計。資料的輸入格式和第三題一模一樣。你的程式必須在讀入這些每筆交易時，將之對應到有在銷售的某種水果，然後對這些水果個別地統計出其營業額和退款額的總和，並依照輸入資料中各水果出現的順序依序印出。不論這些數字各是整數還是小數，一律印出兩位小數。請以不含負號和小數點以外符號的純數字方式印出。兩個數字之間以一個空白鍵隔開。

舉例來說，如果輸入是

```
3 6
apple
banana
orange
Apple $10.0
Banana 10
APPLE 12.1
BANANA -$2.1
apple 2,000.00
banana ($1,000.50)
```

則六個交易額或退款額依序是 10、10、12.1、-2.1、2000 和 -1000.5，其中第一、三、五筆對應到 apple，第二、四、六筆則對應到 banana，因此輸出應該是

```
2022.10 -992.60 0.00
```

你不需要檢查數字的合理性（例如退款總額是否高於銷售總額）。

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。前 15 分由 15 筆測試資料判定分數，一筆測試資料佔 1 分；後 5 分由 5「組」測試資料判定分數，每一組裡面有若干筆測試資料，全對的話才能得到 1 分。

第五題 (bonus)

(20 分) 承第四題，但現在在把交易記錄中代表水果的字串的字母都轉成小寫之後，不一定會跟在第二列到第 $n + 1$ 列定義的水果名稱完全一致（畢竟記帳的人有時也會手誤）。好消息是，交易記錄中的水

果名稱和正確名稱最多只會「差一個字母」，而差一個字母一定是下列三種可能性的其中一種：

1. 多一個字母：例如正確名稱是「apple」，而交易記錄裡寫著「applee」或「appple」。
2. 少一個字母：例如正確名稱是「banana」，而交易記錄裡寫著「banna」或「anana」。
3. 有一個字母被換成另一個字母：例如正確名稱是「orange」，而交易記錄裡寫著「orangg」。

為了避免大家擔心有兩種水果的名稱過於接近，在本題中我們固定 $n = 5$ ，而五種水果的名稱為「apple」、「banana」、「orange」、「watermelon」和「grapefruit」。

你的程式必須在讀入這些說不定有錯的水果名稱時，將之對應到正確的水果名稱，然後對這五種水果個別地統計出其營業額和退款額的總和，並依照上一段列出的順序依序印出。不論這五個數字各是整數還是小數，一律印出兩位小數。請以不含負號和小數點以外符號的純數字方式印出。兩個數字之間以一個空白鍵隔開。

舉例來說，如果輸入是

```
5 6
apple
banana
orange
watermelon
grapefruit
pple $10.0
anana 10
applee 12.1
pbanana -$2.1
apqle 2,000.00
bamana ($1,000.50)
```

則六個交易額或退款額依序是 10、10、12.1、-2.1、2000 和 -1000.5，其中第一、三、五筆對應到 apple，第二、四、六筆則對應到 banana，因此輸出應該是

```
2022.10 -992.60 0.00 0.00 0.00
```

你不需要檢查數字的合理性（例如退款總額是否高於銷售總額）。

針對這個題目，你**可以**使用任何方法。這一題的 40 分都根據程式運算的正確性給分。前 15 分由 15 筆測試資料判定分數，一筆測試資料佔 1 分；後 5 分由 5「組」測試資料判定分數，每一組裡面有若干筆測試資料，全對的話才能得到 1 分。