

程式設計 (106-1)

作業四

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 10 月 10 日凌晨一點**。在你開始前，請閱讀課本的第 6.1-6.4 節¹。第 6.7 和 6.8 節也有幫助。如果你想知道在 C++ 裡面的「函數」是什麼，你可以先讀讀我們下週會教的第五章 (的最前面幾頁)。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) 請回答下列問題：

(a) (10 分) 請解釋下方程式為什麼會印出它所印出的結果。

```
int i = 0;
short sGood = 32765;

while(i < 10)
{
    short sBad = sGood + i;
    cout << sGood + i << " " << sBad << "\n";
    i = i + 1;
}
```

(b) (10 分) 請解釋下方程式有什麼問題、為什麼，以及理論上應該如何修正 (你只需要說明應該如何修正就好，不需要把給定的程式碼修正成理想的程式碼)。

```
int x = 0;
cin >> x;
int array[x];
```

第二題

(20 分) 請回答下列問題：

(a) (10 分) 請解釋下方程式所印出的每個值的意義。

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

```

int a[2][3] = {0};
cout << a << " " << a[0] << "\n";
cout << a[1] << " " << a + 1 << "\n";
cout << sizeof(a) << " " << sizeof(a[0]) << "\n";

```

(b) (10 分) 你剛學 Python 的朋友聽他老師說，Python 裡面有一種叫做 list (清單) 的資料結構，裡面的東西排成一列，每個都有編號，而且編號是從 0、1、2 這樣往下編的。他不知道為什麼編號要從 0 開始。請解釋為什麼 C++ 的 array 編號從 0 開始，給他當個參考。

第三題

(60 分) 在作業三，我們實作了一個 (S, s) 存貨政策。給定一個 (S, s) 存貨政策、相關成本、起始存貨資訊，以及一連串的逐日需求後，我們能計算出總共的缺貨日數、無缺貨日數、總成本等資訊。

很自然地，我們會想問一個問題：最佳的 (S, s) 政策是什麼？也就是說，我們想要找出 S 和 s 的值，以使該 (S, s) 政策下的總成本可以最小化。雖然我們的目標是最小化未來的預期總成本，但我們畢竟不知道未來的需求會長怎樣。一個合理（雖然不完美）的替代方案是：我們來找對過去的需求記錄最佳的 (S, s) ，然後就在未來使用它²。

在本題中，你將被給定 S 的值，然後你應該求出最佳的 s 的值。我們限制 S 和 s 的值都只能是整數，且 $0 \leq s < S$ 。針對每一個可能的 s 值，你應該去計算若我們施行這個 (S, s) 政策，會得到多高的總成本，然後挑出讓總成本最小的 s 值。如果有數個最佳的 s 值，我們就挑最小的那個。

讓我們重新檢視作業三的那個例子。假設 $S = 20$ 、 $s = 5$ 、 $c_o = 2$ 、 $c_u = 5$ 、一開始有 $I_0 = 15$ 個存貨，而連續十天的需求量如表 1 第二列所示，則當日結束時的存貨、當日訂購量、當日存貨成本和當日缺貨成本就會分別被計算如第三、四、五、六列所示。總成本為 205。那如果 $s = 4$ 呢？如表 2 所示， $(20, 4)$ 的存貨政策的總成本是 192，比 $(20, 5)$ 好！事實上，若我們試試 $s = 6$ ，會發現總成本為 172（如表 3 所示）。如果我們嘗試完 0、1、2 直到 19 的 s 值，我們就可以知道哪個 s 值最好。在這個例子裡，最佳的 s 值為 0，總成本為 169³。

i	0	1	2	3	4	5	6	7	8	9	10
x_i	N/A	7	2	12	3	6	6	4	2	6	9
期末存貨	15	8	6	-6	17	11	5	16	14	8	-1
訂貨量	0	0	0	26	0	0	15	0	0	0	21
存貨成本	N/A	16	12	0	34	22	10	32	28	16	0
缺貨成本	N/A	0	0	30	0	0	0	0	0	0	5

表 1: 成本計算範例： $s = 5$

²我們顯然要假設未來的隨機需求的模式會和過往一樣，才會使用根據過去需求求得的 (S, s) 。當然我們可以相信未來的隨機需求模式會改變（例如我們可能相信整個市場需求正處在向上的趨勢中），並據此對求得的 (S, s) 做人為調整，但我們總是要先有一組算出來的 (S, s) ，然後才能做調整。

³ $s = 0$ 有很奇怪嗎？其實還好，就只是「發生缺貨我才訂貨」而已，當缺貨成本不夠大，這也很合理。當然只用 10 筆歷史資料求出的「最佳」 s 值是不太能說服人，但若你能拿出 1000 筆、10000 筆歷史資料，並且確認歷史資料與未來需求間有足夠相關性，那就可行得多了。

i	0	1	2	3	4	5	6	7	8	9	10
x_i	N/A	7	2	12	3	6	6	4	2	6	9
期末存貨	15	8	6	-6	17	11	5	1	18	12	3
訂貨量	0	0	0	26	0	0	0	19	0	0	17
存貨成本	N/A	16	12	0	34	22	10	2	36	24	6
缺貨成本	N/A	0	0	30	0	0	0	0	0	0	0

表 2: 成本計算範例： $s = 4$

i	0	1	2	3	4	5	6	7	8	9	10
x_i	N/A	7	2	12	3	6	6	4	2	6	9
期末存貨	15	8	6	8	5	14	8	4	18	12	3
訂貨量	0	0	14	0	15	0	0	16	0	0	17
存貨成本	N/A	16	12	16	10	28	16	8	36	24	6
缺貨成本	N/A	0	0	0	0	0	0	0	0	0	0

表 3: 成本計算範例： $s = 6$

在本題中，你將被給定 S 與其他相關資訊。請根據本題的規則，求出最佳的 s 值，並且印出該值與該值對應到的總成本。

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。在每個檔案中會有兩行，第一行包含四個整數 S 、 c_o 、 c_u 和 I_0 ，第二行包含 $n + 1$ 個整數，其中第一個整數是 n ，代表後面共有 n 個逐日需求量，第二個數字起則為 x_1 、 x_2 直到 x_n ，其中 x_i 代表第 i 天的單日需求。所有數字都介於 1 和 10000 之間，任意兩個整數之間被一個空白隔開。

讀入資料後，請按照題目的規定，依序印出最佳的 s 值與總成本，任兩個值中間以一個空白字元隔開。舉例來說，如果輸入是

```
20 2 5 15
10 7 2 12 3 6 6 4 2 6 9
```

則輸出應該是

```
0 169
```

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 分會根據你所寫的程式的品質來給分。助教會打開你的程式碼並檢閱你的程式的運算邏輯、可讀性，以及可擴充性（順便檢查你有沒有使用上課沒教過的語法，並且抓抓抄襲）。請寫一個「好」的程式吧！

第四題 (bonus)

(20 分) 承上題，現在我們不僅想要最佳化 s ，也想要最佳化 S ：再給定一個 S 的上限 \bar{S} 後，我們要在 $[1, \bar{S}]$ 之間搜尋 S 的值，並在 $[0, S)$ 之間搜尋 s 的值，以找出使成本最小化的 (S, s) 。如果有複數組 (S, s) 都最小化總成本，我們選 S 跟 s 距離最近的 ($S - s$ 最小的)；還是平手的話，我們選 s 最大的。

本題的輸入格式和第三題一模一樣，唯一的差別是現在輸入的第一個值是 \bar{S} 而非 S 。讀入資料後，請按照題目的規定，依序印出最佳的 S 值、 s 值與總成本，任兩個值中間以一個空白字元隔開。舉例來說，如果輸入是

```
3 2 5 15
10 7 2 12 3 6 6 4 2 6 9
```

則輸出應該是

```
3 2 140
```

請注意 $(S, s) = (3, 1)$ 也會得到總成本 140，此時因為我們要選 $S - s$ 小的，所以我們選 $(S, s) = (3, 2)$ 。

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 1 分。