

程式設計（107-1）

作業五

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 **10 月 30 日凌晨一點**。在你開始前，請閱讀課本的第 5.20–5.22 和第 19 章¹。為這份作業設計測試資料並且提供解答的助教是楊其恆。

第一題

(20 分) 請看 https://en.wikipedia.org/wiki/Catalan_number，去學習關於 Catalan numbers 的相關知識。請寫一個 pseudocode，在給定 $n \geq 0$ 之後，用遞迴的方式找出第 n 個 Catalan number C_n 。舉例來說， $C_0 = 1$ 、 $C_1 = 1$ 、 $C_2 = 2$ 、 $C_3 = 5$ 。

第二題

(20 分) 在作業四第二題，我們用字元陣列實做過大數的加減法。在本題中我們要稍微擴充一下：我們要允許被輸入的大數是負的。更精確地說，我們在本題將要求你做大數（包含正數、負數和零）的加減法，而且這些數字都是金額，因此輸出時要在最前面加上錢字號，並且在正確的地方加入逗號當千分位號。請寫兩個函數

```
void bigMoneyAdd(char m1[], char m2[], char sum[], int n);  
void bigMoneySub(char m1[], char m2[], char diff[], int n);
```

去完成這個任務，其中 $m1$ 和 $m2$ 是兩個代表大整數的字元陣列， sum 和 $diff$ 這兩個字元陣列，分別是裝 $m1 + m2$ 的值或 $m1 - m2$ 的值。 n 是這些陣列的長度（因此你的大數運算依然是有位元限制的，不過至少這個限制可以很寬很寬了）。

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。每個檔案中有三行，其中第一行是一個正整數 n ，去限制等一下會出現的整數及其和與差的位數，第二行和第三行則各是一個整數 m_1 和 m_2 。讀入這些資料後，請先輸出以正確格式帶錢字號和千分位符號的和 $m_1 + m_2$ ，接著輸出以正確格式帶錢字號和千分位符號的差 $m_1 - m_2$ 。如果要輸出的數字是負的，則在該行先輸出錢字號，接著輸出一個負號，在接著輸出該數字（帶千分位符號）的絕對值。已知 $1 \leq n \leq 100$ 、 $-10^n + 1 \leq m_1 \leq 10^n - 1$ 、 $-10^n + 1 \leq m_2 \leq 10^n - 1$ 、 $-10^n + 1 \leq m_1 + m_2 \leq 10^n - 1$ 、 $-10^n + 1 \leq m_1 - m_2 \leq 10^n - 1$ 。

舉例來說，如果輸入是

```
25  
9876543210000000000
```

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

-876123456000000000

則輸出應該是

\$9,000,419,754,000,000,000
\$10,752,666,666,000,000,000

如果輸入是

1
4
-5

則輸出應該是

\$-1
\$9

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。

第三題

(60 分) 你經營一家報攤專賣一份日報，今天下午你得在報社關門前下訂單，告訴報社你要為明天訂購幾份報紙，隔天清晨你就會收到訂購的報紙並且付款。每份報紙的進貨價格是 c 元，賣給客人的零售價則是 r 元。每天會來多少個客人想買報紙是件不確定的事，也就是說單日需求量 D 是隨機的。根據過往經驗，你估計明天的單日需求量會落在 0 和 N 之間，並且符合如下的機率分佈：

$$\Pr(D = i) = p_i, i = 0, 1, \dots, N.$$

意思是說，有 0 個人來買報紙的機率是 p_0 、有 1 個人來買報紙的機率是 p_1 ，依此類推，最後是賣出 N 份報紙的機率是 p_N 。你想要決定你的訂貨量 q^* 去最大化你的期望利潤 (expected profit)

$$\pi(q) = r\mathbb{E}[\min\{q, D\}] - cq,$$

其中 $\min\{q, D\}$ 是明天的銷售量 (訂貨量和需求量中比較小的那個數字)、 $\mathbb{E}[\min\{q, D\}]$ 是預期銷售量 (也就是銷售量取期望值)、 $r\mathbb{E}[\min\{q, D\}]$ 是預期銷售收益、 cq 則是必須付給報社的進貨成本。

以下我們用一個例子來說明這個問題。假設進貨成本 $c = 2$ 、零售價格 $r = 10$ ，而且明天的單日需求量 D 符合如下的機率分佈：

i	0	1	2	3	4	5	6	7	8
$\Pr(D = x_i)$	0.06	0.15	0.22	0.22	0.17	0.10	0.05	0.02	0.01

也就是說有 6% 的機率沒有任何人想買報紙、15% 的機率有一個人想要買，依此類推。你有九個可能的訂貨量 (其實有無限多個，不過你應該不會想訂超過八份報紙吧?)，讓我們來分析其中幾個：

- 如果你不訂貨 ($q = 0$)，顯然一份報紙也不會賣掉，零收益零成本，明天一定賺 0 元。

- 如果你訂一份報紙 ($q = 1$)，
 - 有 6% 的機率會賣掉零份（因為沒人想買），這種情況下你會賺 $0 - 2 = -2$ 元。
 - 有 94% 的機率會賣掉一份（只要有一個以上的人想買），這種情況下你會賺 $10 - 2 = 8$ 元。
 - 你的預期利潤是 $(-2) \times 0.06 + 8 \times 0.94 = 7.4$ 元。
- 如果你訂兩份報紙 ($q = 2$)，
 - 有 6% 的機率會賣掉零份（因為沒人想買），這種情況下你會賺 $0 - 4 = -4$ 元。
 - 有 15% 的機率會賣掉一份（如果恰好有一個人想買），這種情況下你會賺 $10 - 4 = 6$ 元。
 - 有 79% 的機率會賣掉兩份（只要有兩個以上的人想買），這種情況下你會賺 $20 - 4 = 16$ 元。
 - 你的預期利潤是 $(-4) \times 0.06 + 6 \times 0.15 + 16 \times 0.79 = 13.3$ 元。

顯然訂零份或訂一份都不是最好的，因為訂兩份比它們都好。事實上如果你繼續算下去，把 $q = 3, 4, \dots, 8$ 都算過一遍，會發現訂四份是最佳策略 ($q^* = 4$)。當然如果環境改變 (c 和 r 的值或 D 的分佈改變)，結果就可能會不同。

在這題中，你將會被給定 c 、 r 、 N 、 p_0 、 p_1 直到 p_N 的值，你的任務是計算出最佳訂貨量 q^* ，以及在此訂貨量之下能得到的預期利潤 $\pi(q^*)$ 無條件捨去到整數位。以上面的例子而言，就是 4 跟 18（請自己試著算算看）。如果有數個訂貨量會導致一模一樣的預期利潤（是預期利潤一樣，不是無條件捨去之後一樣！），請用比較小的那一個當最佳訂貨量。

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，會有兩列，第一列含有三個整數，依序是單位進貨成本 c 、單位零售價格 r ，以及是需求的上限 N 。第二列含有 $N + 1$ 個小數，依序是賣出零份、一份直到 N 份報紙的機率，同一列的任兩個數字之間被一個空白字元隔開。 c 會落在 1 到 100 之間（包含 1 跟 100）、 r 會落在 1 到 100 之間（包含 1 跟 100）、 r 不會比 c 小、 N 一定會是 8。此外，對於 $i = 0, 1, \dots, N$ ， p_i 會介於 0 到 1 之間（包含 0 跟 1）、最多只有兩位小數，並且滿足 $\sum_{i=0}^N p_i = 1$ 。

讀入這些資料之後，你會計算最佳訂購量 q^* ，以及在此訂購量下的預期利潤無條件捨去到整數 $\lfloor \pi(q^*) \rfloor$ ，並且在兩者中間用一個空格隔開。

舉例來說，如果輸入是

2	10	8						
0.06	0.15	0.22	0.22	0.17	0.1	0.05	0.02	0.01

則輸出應該是

4	18
---	----

其中最佳訂貨量下的預期利潤是 18.5，無條件捨去至 18。

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，我們還是要請你針對給定的問題求取最佳訂貨量，不過問題變得複雜一點。首先， N 會落在 1 到 10000000 之間（包含 1 跟 10000000），而不是固定在 8。其次，明天結束之後，每一份沒賣出去的報紙，可以被以一份 s 元的殘值 (salvage value) 當作廢紙賣掉，也就是說你的預期利潤變成

$$\pi(q) = r\mathbb{E}[\min\{q, D\}] - cq + s\mathbb{E}[\max\{q - D, 0\}],$$

其中 $\max\{q - D, 0\}$ 是沒賣掉的份數 ($D \geq q$ 表示沒有剩)，因此 $s\mathbb{E}[\max\{q - D, 0\}]$ 是預期總殘值。

本題如果要做 linear search 把所有 0 到 N 之間的整數都試一遍，對於每一個可以訂購的量，再用一個迴圈去把所有可能的需求值跑一遍去算銷售量，以如此的雙層迴圈結構為你的演算法，那算是不太難。但如果 N 很大，這麼做就會很浪費時間。請自行思考這個問題，找出最佳解的性質（舉例來說，如無意外應該不會是 0 或 N ），並且利用你找到的性質來寫一個比較好的演算法。如果你只能寫出基於 linear search 的雙層迴圈，說不定會一分都拿不到。

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，會有兩列，第一列含有四個整數，依序是單位進貨成本 c 、單位零售價格 r 、需求的上限 N ，以及單位殘值 s 。第二列含有 $N + 1$ 個小數，依序是賣出零份、一份直到 N 份報紙的機率，同一列的任兩個數字之間被一個空白字元隔開。

輸出的要求與第三題相同。舉例來說，如果輸入是

2 10 10 1
0.06 0.15 0.22 0.22 0.17 0.1 0.05 0.02 0.01 0 0

則輸出應該是

5 20

其中最佳訂貨量下的預期利潤是 20.47，無條件捨去至 20。

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。