**全國高級中等學校106學年度工業類科學生技藝競賽電腦軟體設計**

**壹、試卷說明:**

**1.請將寫好之程式原始檔依題號命名資料夾存檔，第一題取姓名\_Q1，第二題取姓名\_Q2，依序命名存檔，**

**並存於C碟之資料夾”姓名\_Contest”中。**

**2.競賽時間4小時。**

**3將程式及編譯成執行檔儲存在C碟之資料夾姓名\_Contest。**

**貳、評分說明:本試卷共六題，每題配分不一。**

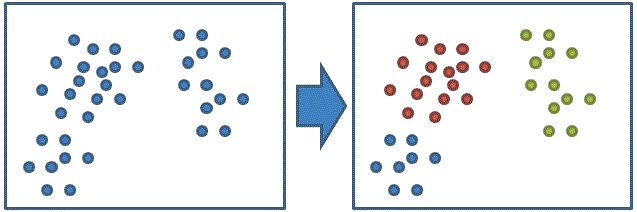
**1. 每題評分只有對與錯兩種，對則給滿分，錯則不給分(即以零分計算) 。**

**2. 每解答完一題上傳(程式及執行檔)，評審人員將針對該題進行測試，若解題正確則回應正確，若解題錯誤則扣該題一分至該題零分為止，答錯之題目可繼續作答。**

**試題1：資料分群(17分)**

**說明:當有大量資料，想將資料量大大地縮小時，有很多方法，其中之一是聚類法或分群法。基本上就是將有相似的特徵分在同一群，也就是照著物以類聚的方式在進行分群，最後以各群(類)的重心作為代表。**

**如左下圖給予N=36筆資料，將之分為k類(k=3)如右下圖 (k由使用者設定)**



**此聚類法的目標是使下列公式*Mean Squared Error (MSE)*達到最小化**

**所有各群資料點 到其對應群重心 的距離總合是最小的，也就是要找到最佳的群重心  及 所屬的群來符合上面的要求(為第i群)。演算法如下:**

**S0:產生N筆資料，設定要將資料分類為k 群及重複疊代的次數Q；初值P=0**

**S1: 從N筆資料中隨機選取的k筆資料當作初始群重心~**

**S2: While (P<Q) Do**

**S3: 將每一筆資料分別與各個群重心分別計算其歐式距離，比較每筆資料和那一個群重心距離最短，就分配該筆資料至距離最短的那一群。例如每筆資料有兩個值=(,)，群重心=(，)，歐式距離為:**

**S4: 利用得到的分群結果重新計算中心點(分別對**每個**群裡的資料，重新計算平均值** 作為新的**重心)**

**S5:計算第P次MSE**

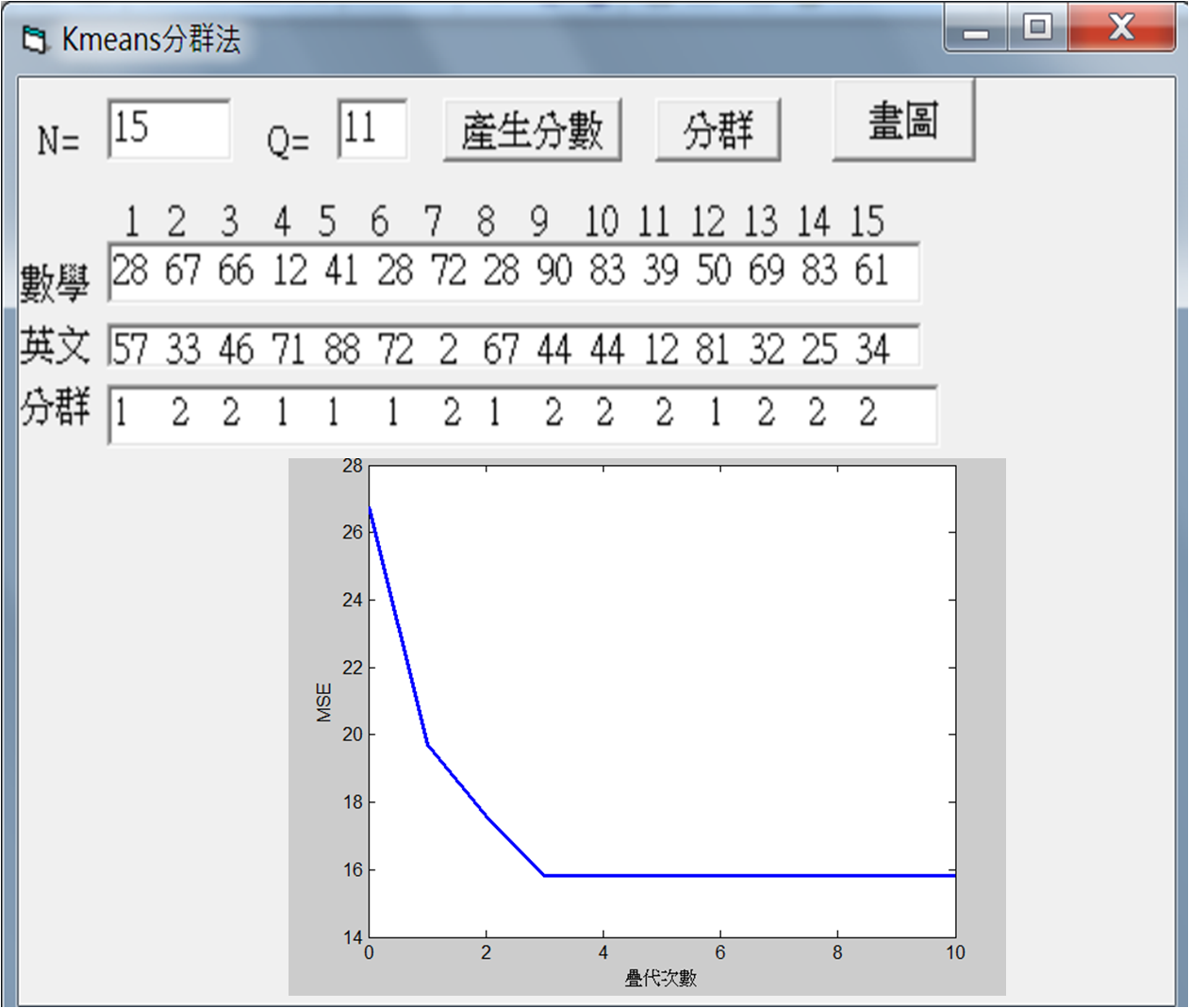
SE=

**S5: P=P+1;**

**S6: end while**

**為了簡化起見，設一班學生人數N可有15至30人，依每人的資料為數學和英文兩科成績，按上述演算法將全班分為兩群。數學和英文兩科成績可藉由隨機產生，分數介於0到100 之間，可設定Q介於11到20 之間。請設計一程式輸入學生人數N及疊代的次數Q，隨機產生N個數學和英文兩科成績，分群結束後列出各個學生分群結果及最後畫出每次重複疊代的MSE值如下圖。**

**例子:如下圖輸入N=15, Q=11,按下"產生分數鍵"隨機產生數學和英文兩科成績；接著按下"分群鍵"可列出分群結果；再按"畫圖"可畫出第0~Q-1次分群後所得到的MSE值。**

**

**試題2：簡易依序循環之排程 (Round-Robin Scheduling) 程式(17分)**

**說明：**在分時作業系統中，排程程式會設定一個**時間配額** (time quantum)，然後依序輪流執行就緒狀態中的每個**行程** (process)。如果執行中的行程在未用完時間配額前，就完成了工作，就會自動交還CPU 的執行權，給下一個行程執行；如果執行的行程用完了時間配額，卻還沒完成工作，還是得交出CPU 的執行權，改由下一個行程使用。如此循環執行每個就緒狀態中的行程，直到所有行程結束為止。

例如有 3 個正在等待被執行的行程，各個行程所需CPU執行時間如圖2-1所示。

|  |  |
| --- | --- |
| 圖2-1 | 圖2-2 |
| 圖2-3 | |

若時間配額為 4 時間單位，且成為就緒狀態的先後順序為P1、P2、P3 (但幾乎同時到達)，P1 行程是第一個在時間00被執行，如圖2-2所示。P1 行程用完4 時間單位的時間配額後, 雖還未完成工作，但CPU執行權仍會被強制交由P2 行程；P2 行程在時間04被執行只需3時間單位即完成工作，因此還沒有用完時間配額，就結束了行程；接著，排程程式將CPU 執行權交給下一個行程，P3行程在時間07被執行 … 依此類推，直到所有行程完成工作為止，如圖2-3所示。

P1、P2、P3等待時間分別為12(=3+4+4+1)、4、15(=4+3+4+4)時間單位。

**目標：**

1. 請寫一支程式如圖2-3、圖2-4所示。在圖2-4中，輸入行程processes數量為5，  
   每個行程的執行時間burst\_time，分別為10、5、15、5、10，  
   接著輸入時間配額time\_quantum為5，  
   您寫的程式可以將所有行程processes執行順序及等待時間顯示出來，如圖2-4所示。

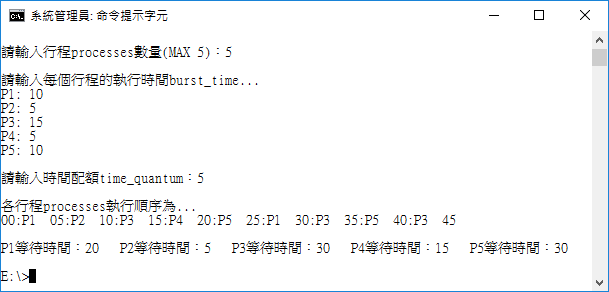


圖2-4

1. 若輸入的行程processes數量大於5，則您寫的程式需重新要求輸入行程processes數量，  
   如圖2-5所示。

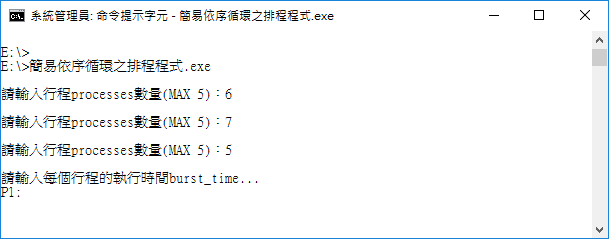


圖2-5

**試題3：辨認數字系統(Recognized Number System) (17分)**

**說明：**(一)已知一個七段顯示器能呈現各種樣式，辨認數字系統能辨認它的顯示樣式為**非數字**或為如下圖所示**數字0至9之一**，其中數字1、6與9各有兩種顯示樣式。



**(二)** 請設計如下圖所示之「**簡易辨認數字系統**」，每當滑鼠左鍵點一下**Random Set**鍵，該系統的中間七段顯示器之各段會隨意產生ON (亮) 或OFF (不亮) 而呈現一種樣式，同時系統啟動自動辨認，如果此顯示樣式(**假設為6**)為數字0至9之一，則辨認 **數字為： 6** ；否則自動辨認 **數字為：非數字**。



**(三)** 該系統還具有手工對七段顯示器隨意設定各段ON (亮) 或OFF (不亮) 而改變其顯示樣式，即每當使用滑鼠左鍵對任一段點一下，如此段原為ON則更改為OFF，原為OFF則更改為ON。進一步，每當滑鼠點一下**Recognizing**鍵，則系統自動辨認目前顯示樣式之 **數字為： 3 (假設樣式為3)**或 **非數字**。

**(四)** 上述可重複操作，直至以滑鼠左鍵點一下**Exit**鍵，則離開該系統。

**範例1**

每當滑鼠點一下該系統之**Random Set**鍵，則七段顯示器隨意產生一個樣式，且系統自動辨認此顯示樣式之 **數字為：非數字** 或 **8** (**假設樣式為8**) 或 **6** (**假設樣式為6**)。

   **……**

**範例2**

已知系統對七段顯示器隨意產生一個樣式如左下圖，並以手工使用滑鼠左鍵對七段修改為如右下圖之樣式(**假設為7**)，接著每當滑鼠點一下**Recognizing**鍵，則系統自動辨認目前顯示樣式之 **數字為： 7** 。

**試題4：小明的機器人(17分)**

**說明**：小明想要設計一自動左右轉的機器人，此機器人可以接受輸入命令，然後得到輸出指令:向左和向右。為了達到此目的，小明事先收集一些輸入命令和輸出指令對資料，來訓練機器人，等訓練完畢後，機器人可以接受輸入命令，自行判斷得到輸出指令，要向左還是向右。為了達到上述目的，小明利用下列式子來讓機器人學習:

 (1)

其中*x* = (*x1*, *x2*,…, *xn*)是*n*個輸入命令，*w* = (*w1*, *w2*,…, *wn*)分別是*n*個輸入命令*x* = (*x1*, *x2*,…, *xn*)的權重。*f*是公式(1)計算得到的數值，為了得到機器人的兩種輸出指令*o*: 1代表向左, -1代表向右，利用下列式子，來將式子(1)計算得到的數值轉為兩種輸出指令。

 (2)

sign是符號判斷函數，當*f*是正數時，*o*為1，當*f*是負數時，*o*為-1。

從式子(1)可知，只要權重*w* = (*w1*, *w2*,…, *wn*)知道，機器人就可以利用式子(1)和(2)，來決定要向左還是要向右的命令。為了學習權重*w* = (*w1*, *w2*,…, *wn*)，每次訓練後，利用下面式子來進行權重更新:

 (3)

其中符號*η*是學習率。為了完成上述訓練工作，小明用以下方法來達成:

Step 1. 收集*K*筆訓練用的輸入命令和輸出指令對資料(*x1*, *y1*),…,(*xK*, *yK*)，其中(*x1*,…,*xK*)為輸入命令, (*y1*,…,*yK*)為輸出指令。

Step 2. 學習率初始為*η* > 0，初始權重*w*，初始累積錯誤*E*=0，*k*=1(即從第一筆資料開始訓練)。

Step 3. 開始訓練: 輸入命令是*xk*，帶入公式(1)，此時的*f*為*fk* = <*wk-1*, *xk*>，再利用公式(2)，可以求輸出指令*o*(*xk*) = sign(*fk*) = sign(<*wk-1*, *xk*>)

Step 4. 更新權重:  (4)

Step 5. 利用下面公式來計算累積錯誤*E* =*E* + 0.5 \* ||*yk*- *o*(*xk*)||2

Step 6. 當還有訓練的輸入命令和輸出指令對時，繼續下一筆資料對的訓練，即當*k*<*K*時，*k*=*k*+1，回到Step 3繼續訓練。否則跳到Step 7。

Step 7. 訓練週期結束。當累積錯誤*E*=0，就結束訓練。若是累積錯誤*E*>0，就*E*=0，*k*=1，跳到Step 3，啟動新的訓練週期。

**範例:**

假設收集3筆訓練用的輸入命令和輸出指令資料對如下:

|  |  |  |
| --- | --- | --- |
| 筆 | 輸入命令*x* | 輸出指令*y* |
| 1 | *x1* = (1, 0, 1) | *y1* = -1 |
| 2 | *x2* = (0, -1, -1) | *y2* = 1 |
| 3 | *x3* = (-1, -0.5, -1) | *y3* = 1 |

每一筆輸入*x*有3個命令，初始學習率*η* = 0.1，初始權重*w*0 = (1, -1, 0)，依照上述訓練方法，其過程如下:

1. 第一筆訓練的輸入命令為*x*1 = (1, 0, 1)，其輸出指令為*y*1 = -1，帶入公式(1)，此時的*f*為

*f1* = <*w0*, *x1*> = (1)\*(1) + (-1)\*(0) + (0)\*(1) = 1，再利用公式(2)，

求輸出指令*o*(*x1*) = sign(*f1*) = sign(1) = 1，此結果不等於*y*1 = -1，要用公式(4)來更新權重:

*w1* = *w*0 + 0.1(-1 – (1))*x1* = *w*0 - 0.2*x1*=

累積錯誤*E* = *E* + 0.5 \* ||*y1*- *o*(*x1*)||2 = 0 + 0.5 \* ||-1 – (1)||2 = 0 + 0.5 \* ||-2||2 = 0.5\*4 = 2。

1. 第二筆訓練的輸入命令為*x*2 = (0, -1, -1)，其輸出指令為*y*2 = 1，帶入公式(1)，此時*f*為

*f2* = <*w1*, *x2*> = (0.8)\*(0) + (-1)\*(-1) + (-0.2)\*(-1) = 1.2，再利用公式(2)，

求輸出*o*(*x2*) = sign(*f2*) = sign(1.2) = 1，此結果等於*y*2 = 1，所以權重不用更新，即*w2* = *w1*。

累積錯誤*E* = *E* + 0.5 \* ||*y2*- *o*(*x2*)||2 = 2 + 0.5 \* ||1 – 1||2 = 2。

1. 第三筆訓練的輸入命令為*x*3 = (-1, -0.5, -1)，其輸出指令為*y*3 = 1，帶入公式(1)，此時*f*為

*f3* = <*w2*, *x3*> = (0.8)\*(-1) + (-1)\*(-0.5) + (-0.2)\*(-1) = -0.1，再利用公式(2)，

求輸出*o*(*x3*) = sign(*f3*) = sign(-0.1) = -1，此結果不等於*y*3= 1，要用公式(4)來更新權重:

*w3* = *w*2 + 0.1(1 – (-1))*x3*

= *w*2 + 0.2*x3* =

累積錯誤*E* = *E* + 0.5 \* ||*y3*- *o*(*x3*)||2 = 2 + 0.5 \* ||1 – (-1)||2 = 2 + 0.5\*||2||2 = 2 + 0.5\*4 = 4。

1. 累積錯誤*E* > 0，*E* = 0，*k* = 1，跳到Step 3，啟動新的訓練週期。

第一筆訓練的輸入命令為*x*1 = (1, 0, 1)，其輸出指令為*y*1 = -1，帶入公式(1)，此時*f*為

*f1* = <*w3*, *x1*> = 0.6\*(1) + (-1.1)\*(0) + (-0.4)\*(1) = 0.2，再利用公式(2)，

求輸出*o*(*x1*) = sign(*f1*) = sign(0.2) = 1，此結果不等於*y* 1= -1，要用公式(4)來更新權重:

*w4* = *w*3 + 0.1(-1 – (1))*x1*

= *w*3 – 0.2*x1*=

累積錯誤*E* = *E* + 0.5 \* ||*y1*- *o*(*x1*)||2 = 0 + 0.5 \* ||-1 – (1)||2 = 0 + 0.5\*||-2||2 = 0.5\*4 = 2。

1. 第二筆訓練的輸入命令為*x*2 = (0, -1, -1)，其輸出指令為*y*2 = 1，帶入公式(1)，此時*f*為

*f2* = <*w4*, *x2*> = 0.4\*(0) + (-1.1)\*(-1) + (-0.6)\*(-1)=1.7，再利用公式(2)，

求輸出*o*(*x2*) = sign(*f2*) = sign(1.7) = 1，此結果等於*y*2 = 1，權重不用更新，即*w5* = *w4*。

累積錯誤*E* = *E* + 0.5 \* ||*y2*- *o*(*x2*)||2 = 2 + 0.5 \* ||1 – 1||2 = 2。

1. 第三筆訓練的輸入命令為*x*3 = (-1, -0.5, -1)，其輸出指令為*y*3 = 1，帶入公式(1)，此時*f*為

*f3* = <*w5*, *x3*> = 0.4\*(-1) + (-1.1)\*(-0.5) + (-0.6)\*(-1) = 0.75，再利用公式(2)，

求輸出*o*(*x3*) = sign(*f3*) = sign(0.75) = 1，此結果等於*y*3 = 1，權重不用更新，即*w6*= *w5*。

累積錯誤*E* = *E* + 0.5 \* ||*y3*- *o*(*x3*)||2 = 2 + 0.5 \* ||1 – 1||2 = 2。

1. 累積錯誤*E* > 0，*E* = 0，*k* = 1，跳到Step 3，啟動新的訓練週期。

<*w6*, *x1*>=(0.4)\*(1) + (-1.1)\*(0) + (-0.6)\*(1) = -0.2 < 0，即sign(-0.2) = -1 = *y*1 = -1，*E = 0。*

<*w6*, *x2*>=(0.4)\*(0) + (-1.1)\*(-1) + (-0.6)\*(-1) = 1.7 > 0，即sign(1.7) = 1 = *y*2 = 1，*E = 0。*

<*w6*, *x3*>=(0.4)\*(-1) + (-1.1)\*(-0.5) + (-0.6)\*(-1) = 0.75 > 0，即sign(0.75) = 1 = *y*3 = 1，*E = 0。*

累積錯誤***E* = 0**，結束訓練，最後得到的權重為***w6* = (0.4, -1.1, -0.6)**。

1. 再來利用訓練得到的權重，來測試新的輸入命令*x4* = (1, 1, 1)，帶入公式(1)，此時的*f*為

*f4* = <*w6*, *x4*> = (0.4)\*(1) + (-1.1)\*(1) + (-0.6)\*(1) = -1.3，再利用公式(2)，

求輸出*o*(*x4*) = sign(*f4*) = sign(-1.3) = -1，輸出指令為-1，即機器人要向右。

**程式功能**：

請利用上述說明和範例，寫一個程式，能完成以下功能要求:

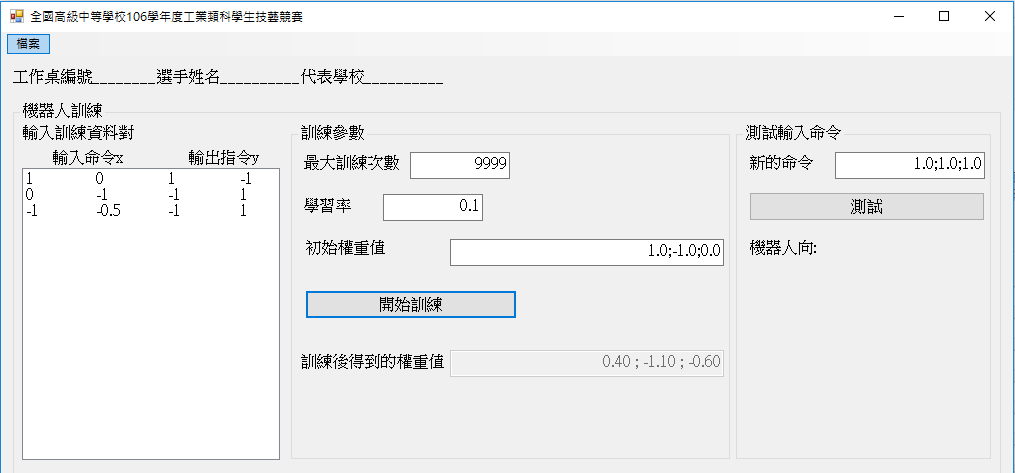
1. 能讓使用者選擇輸入訓練資料對檔案並顯示出來。
2. 能讓使用者輸入<訓練參數>並按<開始訓練>，顯示正確的訓練後權重值。
3. 能讓使用者輸入<測試輸入命令>並按<測試>，顯示正確的輸出指令。

**程式執行範例**：

1. 使用者選擇輸入訓練資料對檔案並顯示出來。此檔案有3筆資料。每一筆都有3個輸入命令和1個輸出指令。



(2)使用者輸入<訓練參數>並按<開始訓練>，顯示正確的訓練後權重值(**0.40; -1.10; -0.60)**。



(3)使用者輸入<測試輸入命令>並按<測試>，可以顯示正確的輸出指令:**機器人向:-1(右)**。



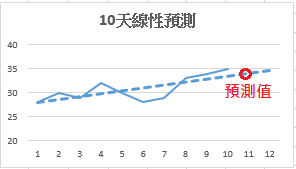
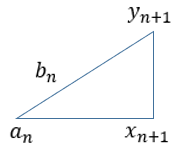
***若妳（你）的程式都完成上述功能和要求，才可以要求檢查功能。***

**試題5：交易價格預測的程式設計(16分)**

**說明：**一、有自由交易活動，就會留下交易資料。當數筆交易價產生之後，後續的交易價如何發展？常常是引起人們興趣的話題。對於價格走勢的預測已有很多研究的發表，而在日常的應用，也可以Microsoft Excel來做預測的處理，例如：有15筆交易價序列28, 30, 29, 32, 30, 28, 29, 33, 34, 35, 34, 32, 33, 35, 34，則短期預測以十天(28, 30, 29, 32, 30, 28, 29, 33, 34, 35)的預測線如圖5-1的虛線所示。由預測線的角度(或直線的斜率)即可表示預測價格可能的發展。

二、然而交易價是多變的，隨著交易價的改變，預測線的角度(或直線的斜率)也會跟著調整，因此，我們需要將這些預測加以**程式化**，在此採用**簡單線性迴歸**，其公式為 Yt = a+bx ，式中x是自變數，是選定的任何x值；Yt是因變數，對於選定的x值，相應變數Y的平均估計值，即第t預測周期的預測值；a、b是迴歸係數。

相關係數r是一個在0和正負1之間的係數，為了簡化問題，在此暫時不計，只用一條直線*y*＝*a*＋*bx* 來描述預測變數隨日期的變動趨勢。



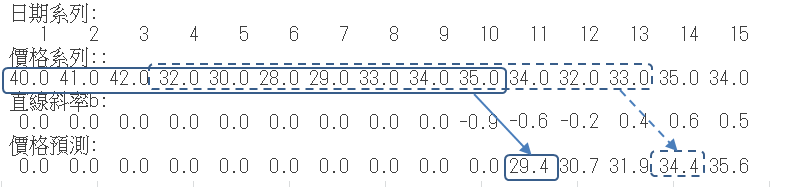
圖5-1 以10天交易價的預測線 圖5-2 預測**次日**的

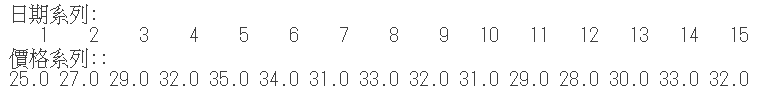
圖5-3 一系列預測值計算的結果

三、對於已交易價序列，利用**最小平方法原理**計算以*n*為週期的值，求出它們的a、b值，

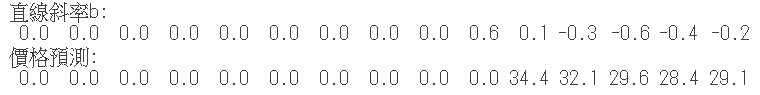
|  |  |  |  |
| --- | --- | --- | --- |
| ■平均值： |  | ■迴歸係數： |  |

其中*x*代表日期序列，以自然數表示，例如：1,2,3,4,5；*y*代表已交易價序列。我們只要求出各a、b值，經由直線*y*＝*a*＋*bx*，即可由已交易價序列計算出預測**次日()**的之變動趨勢，亦即，如圖5-2所示，其中、為某日之迴歸係數，為某日之次日，為某日之次日的交易價。在此，假設週期*n*為10，則一系列預測值的計算結果如圖5-3所示。

**輸入格式：**

****

**輸出格式：**

****

**操作範例1：**

|  |  |
| --- | --- |
| **輸入：** |  |
| **輸出：** |  |

**操作範例2：**

|  |  |
| --- | --- |
| **輸入：** |  |
| **輸出：** |  |

**題目6: n！程式設計(16分)**

1. 說明：**輸入的每一行測試資料有1個整數 n（2<=n <= 100）**
2. **對每一個輸入n，第一行輸出n!的值。例如：8! = 40320。**
3. **對每一個輸入n，第二行輸出n!的每個位數的總和。這個數一定小於231-1。例如：8! = 40320，每個位數的總和為4+0+3+2+0 = 9。**
4. **對每一個輸入n，第三行輸出最多可以用多少整數相乘（1 除外）來表達 n!。換言之，也就是質因數分解後的指數相加總和。**

**例如：8! = 1.2.3.4.5.6.7.8 = 2.3.2.2.5.3.2.7.2.2.2 = 27 .32 .5.7，可以很清楚的觀察到最多可以用 11 個整數（1 除外）相乘來產生 8! 。**

1. 輸入說明:輸入含有多組測試資料。每組測試資料為1個整數 n（2 <= n <= 100 ）。請參考範例:
2. 範例輸入

8

12

20

100

1. 輸出說明

對每組測試資料輸出3行。

第一行輸出n!的值。

第二行輸出n!的每個位數的總和。

第三行輸出最多可以用多少個整數（1 除外）相乘來產生n!。換言之，也就是質因數分解後的指數相加總和。

1. 範例輸出

40320

9

11

479001600

27

19

2432902008176640000

54

36

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000000

648

239

1. 評分標準
2. 介面不拘，可以輸入測資並輸出結果即可。
3. 參考測資有四組，隱藏測資有多組，所有測資皆正確，才有分數。