

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。

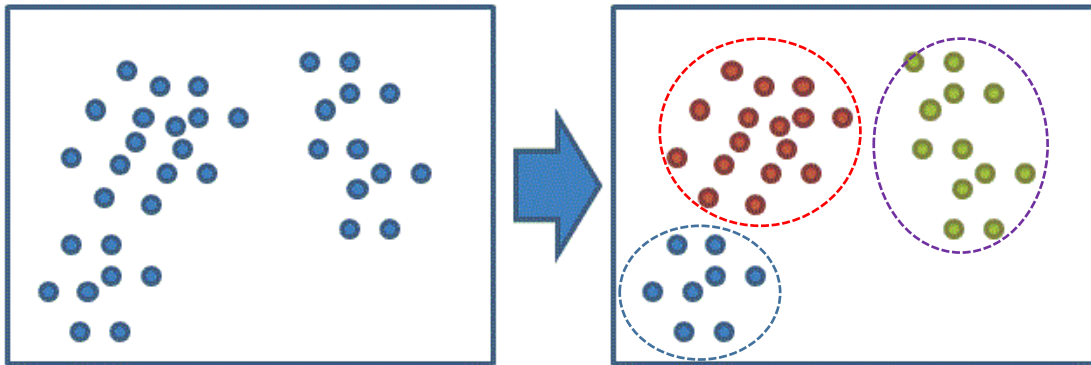
1. 每題評分只有對與錯兩種，對則給滿分，錯則不給分(即以零分計算)。

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

$$MSE = \frac{1}{N} \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

所有各群資料點 x_j 到其對應群重心 μ_i 的距離總合是最小的，也就是要**找到最佳的群重心 μ_i 及 x_j 所屬的群**來符合上面的要求(S_i 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 $\mu_1 \sim \mu_k$

S2: While (P<Q) Do

S3: 將每一筆資料 x_j 分別與各個群重心 μ_j 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 $x_j=(x_{j1}, x_{j2})$ ，群重心 $\mu_i=(\mu_{i1}, \mu_{i2})$ ，歐式距離為：

$$d(x_j, \mu_i) = \sqrt{(x_{j1} - \mu_{i1})^2 + (x_{j2} - \mu_{i2})^2}$$

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 μ_i 作為新的重心)

S5:計算第P次MSE

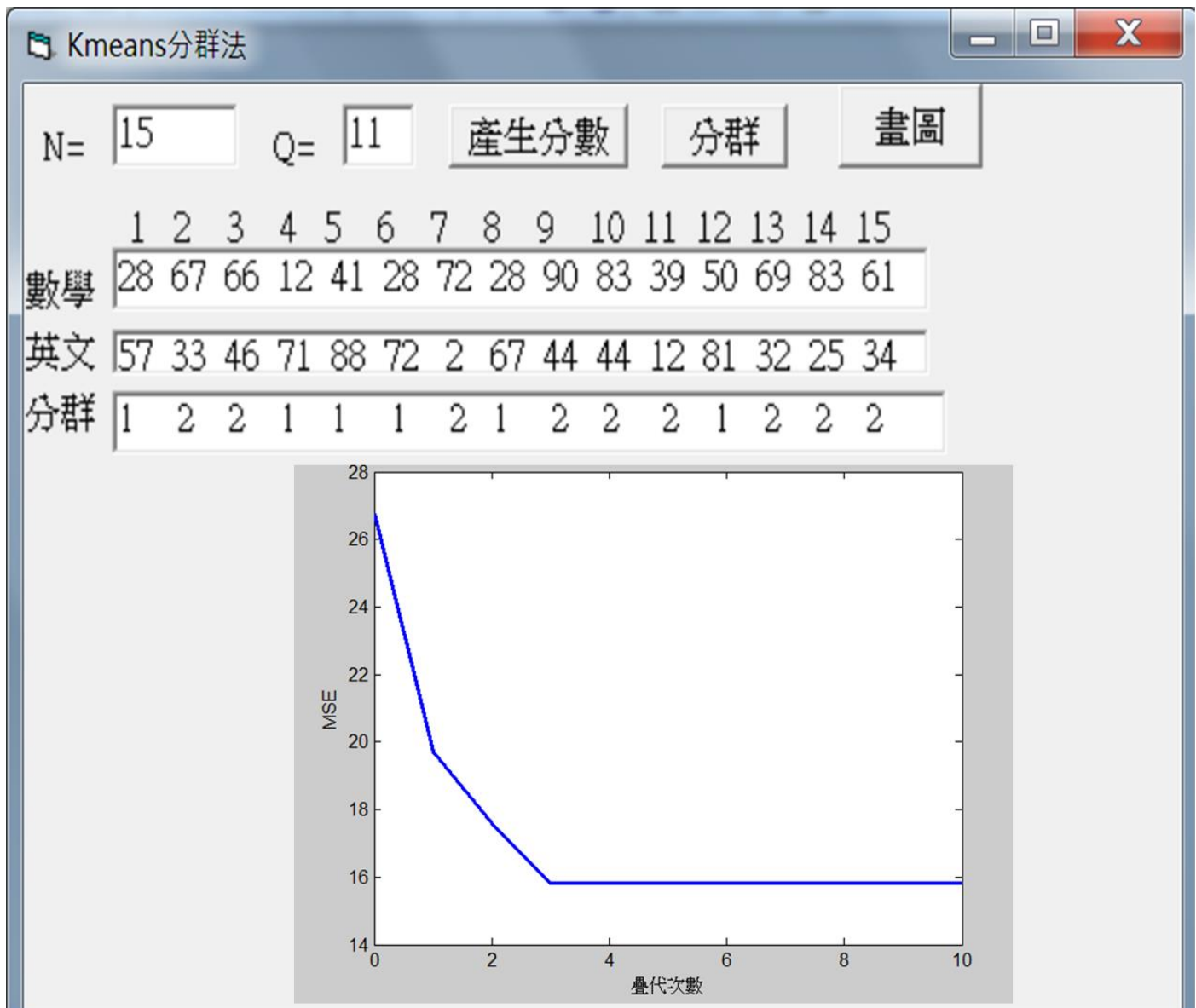
$$MSE = \frac{1}{N} \sum_{i=1}^k \sum_{x_j \in S_i} [(x_{j1} - \mu_{i1})^2 + (x_{j2} - \mu_{i2})^2]$$

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"隨機產生數學和英文兩科成績；接著按下"分群"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。

行程(process)	所需CPU執行時間(burst time)
P1	21
P2	3
P3	9

圖 2-1

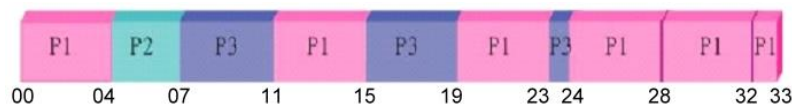


圖 2-2

```

C:\> 選取 系統管理員: 命令提示字元
E:\>
E:\>簡易依序循環之排程式.exe

請輸入行程processes數量(MAX 5) : 3

請輸入每個行程的執行時間burst_time...
P1: 21
P2: 3
P3: 9

請輸入時間配額time_quantum : 4

各行程processes執行順序為...
00:P1 04:P2 07:P3 11:P1 15:P3 19:P1 23:P3 24:P1 28:P1 32:P1 33

P1等待時間 : 12   P2等待時間 : 4   P3等待時間 : 15

E:\>

```

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 時間單位的时间配額後，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。

```

C:\> 系統管理員: 命令提示字元
E:\>
E:\>簡易依序循環之排程式.exe

請輸入行程processes數量(MAX 5) : 5

請輸入每個行程的執行時間burst_time...
P1: 10
P2: 5
P3: 15
P4: 5
P5: 10

請輸入時間配額time_quantum : 5

各行程processes執行順序為...
00:P1 05:P2 10:P3 15:P4 20:P5 25:P1 30:P3 35:P5 40:P3 45

P1等待時間 : 20   P2等待時間 : 5   P3等待時間 : 30   P4等待時間 : 15   P5等待時間 : 30

E:\>

```

圖 2-4

2. 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。

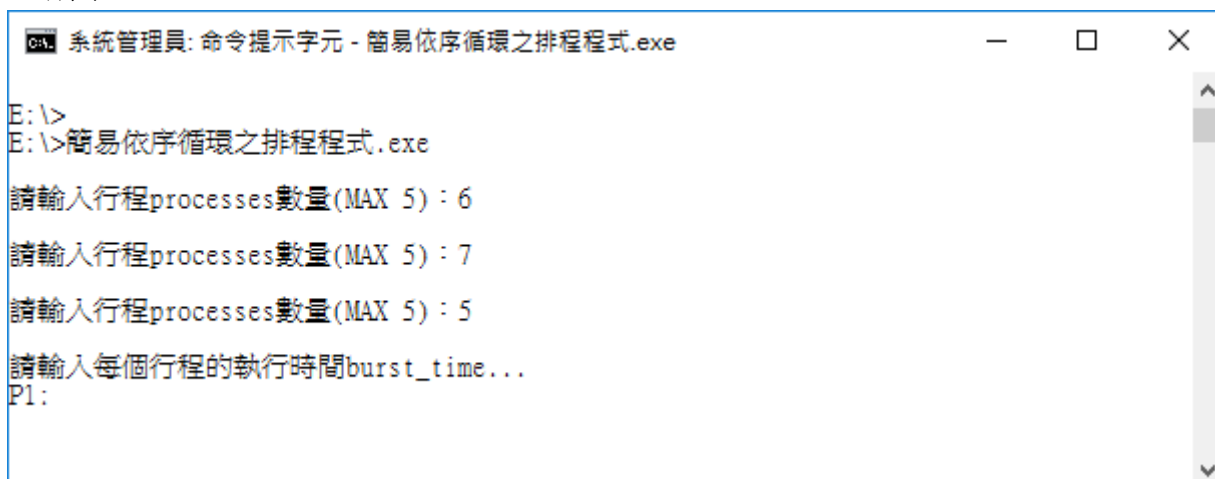
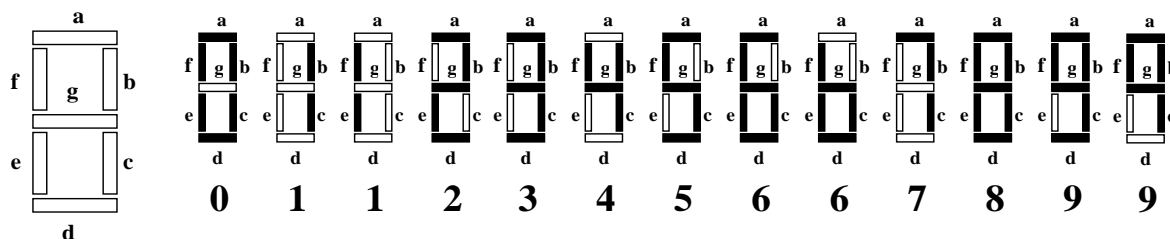


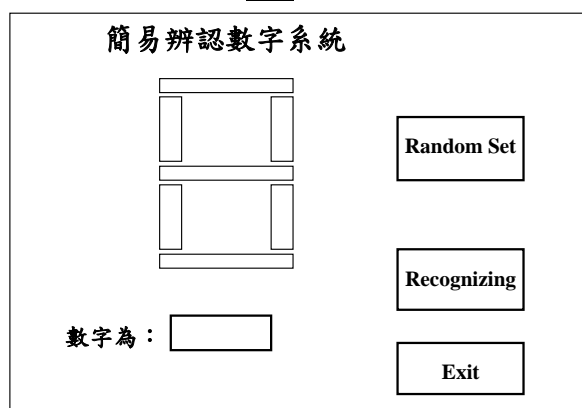
圖 2-5

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**。

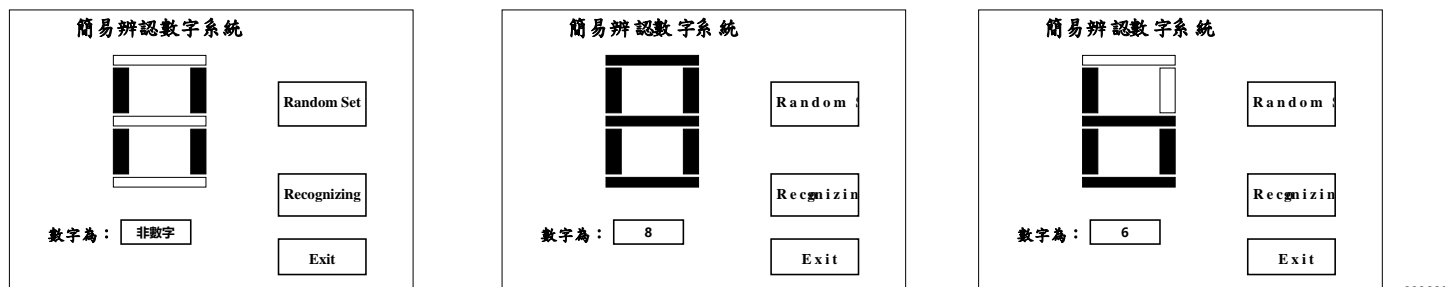


(三) 該系統還具有手工對七段顯示器隨意設定各段 ON(亮) 或 OFF(不亮) 而改變其顯示樣式，即每當使用滑鼠左鍵對任一段點一下，如此段原為 ON 則更改為 OFF，原為 OFF 則更改為 ON。進一步，每當滑鼠點一下 **Recognizing** 鍵，則系統自動辨認目前顯示樣式之 **數字為：3**(假設樣式為 3)或 **非數字**。

(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。

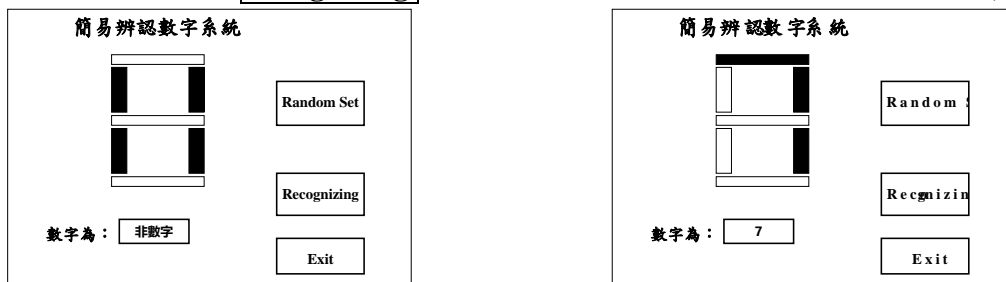
範例 1

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：

$$f = \langle w, x \rangle = w_1x_1 + w_2x_2 + \dots + w_nx_n = \sum_{j=1}^n w_jx_j \quad (1)$$

其中 $x = (x_1, x_2, \dots, x_n)$ 是 n 個輸入命令， $w = (w_1, w_2, \dots, w_n)$ 分別是 n 個輸入命令 $x = (x_1, x_2, \dots, x_n)$ 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。

$$o = \text{sign}(f) = \begin{cases} 1 & \text{if } f \geq 0 \\ -1 & \text{if } f < 0 \end{cases} \quad (2)$$

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 $w = (w_1, w_2, \dots, w_n)$ 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 $w = (w_1, w_2, \dots, w_n)$ ，每次訓練後，利用下面式子來進行權重更新：

$$w = w + \eta(y - \text{sign}(f))x \quad (3)$$

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 $(x^1, y^1), \dots, (x^K, y^K)$ ，其中 (x^1, \dots, x^K) 為輸入命令， (y^1, \dots, y^K) 為輸出指令。

Step 2. 學習率初始為 $\eta > 0$ ，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 x^k ，帶入公式(1)，此時的 f 為 $f^k = \langle w^{k-1}, x^k \rangle$ ，再利用公式(2)，可以求輸出指令 $o(x^k) = \text{sign}(f^k) = \text{sign}(\langle w^{k-1}, x^k \rangle)$

Step 4. 更新權重： $w^k = w^{k-1} + \eta(y^k - \text{sign}(\langle w^{k-1}, x^k \rangle))x^k \quad (4)$

Step 5. 利用下面公式來計算累積錯誤 $E = E + 0.5 * ||y^k - o(x^k)||^2$

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	$x^1 = (1, 0, 1)$	$y^1 = -1$
2	$x^2 = (0, -1, -1)$	$y^2 = 1$
3	$x^3 = (-1, -0.5, -1)$	$y^3 = 1$

每一筆輸入 x 有 3 個命令，初始學習率 $\eta = 0.1$ ，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 $f^1 = \langle w^0, x^1 \rangle = (1)*(1) + (-1)*(0) + (0)*(1) = 1$ ，再利用公式(2)，

求輸出指令 $o(x^1) = \text{sign}(f^1) = \text{sign}(1) = 1$ ，此結果不等於 $y^1 = -1$ ，要用公式(4)來更新權重：

$$w^1 = w^0 + 0.1(-1 - (1))x^1 = w^0 - 0.2x^1 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} - 0.2 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0.2 \\ 0 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 1-0.2 \\ -1-0 \\ 0-0.2 \end{pmatrix} = \begin{pmatrix} 0.8 \\ -1 \\ -0.2 \end{pmatrix}$$

累積錯誤 $E = E + 0.5 * ||y^1 - o(x^1)||^2 = 0 + 0.5 * ||-1 - (1)||^2 = 0 + 0.5 * ||-2||^2 = 0.5 * 4 = 2$ 。

2. 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 $f^2 = \langle w^1, x^2 \rangle = (0.8)*(0) + (-1)*(-1) + (-0.2)*(-1) = 1.2$ ，再利用公式(2)，求輸出 $o(x^2) = \text{sign}(f^2) = \text{sign}(1.2) = 1$ ，此結果等於 $y^2 = 1$ ，所以權重不用更新，即 $w^2 = w^1$ 。

累積錯誤 $E = E + 0.5 * ||y^2 - o(x^2)||^2 = 2 + 0.5 * ||1 - 1||^2 = 2$ 。

3. 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 $f^3 = \langle w^2, x^3 \rangle = (0.8)*(-1) + (-1)*(-0.5) + (-0.2)*(-1) = -0.1$ ，再利用公式(2)，

求輸出 $o(x^3) = \text{sign}(f^3) = \text{sign}(-0.1) = -1$ ，此結果不等於 $y^3 = 1$ ，要用公式(4)來更新權重：

$$w^3 = w^2 + 0.1(1 - (-1))x^3$$

$$= w^2 + 0.2x^3 = \begin{pmatrix} 0.8 \\ -1 \\ -0.2 \end{pmatrix} + 0.2 \begin{pmatrix} -1 \\ -0.5 \\ -1 \end{pmatrix} = \begin{pmatrix} 0.8 \\ -1 \\ -0.2 \end{pmatrix} + \begin{pmatrix} -0.2 \\ -0.1 \\ -0.2 \end{pmatrix} = \begin{pmatrix} 0.8-0.2 \\ -1-0.1 \\ -0.2-0.2 \end{pmatrix} = \begin{pmatrix} 0.6 \\ -1.1 \\ -0.4 \end{pmatrix}$$

累積錯誤 $E = E + 0.5 * ||y^3 - o(x^3)||^2 = 2 + 0.5 * ||1 - (-1)||^2 = 2 + 0.5 * ||2||^2 = 2 + 0.5 * 4 = 4$ 。

4. 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 $f^1 = \langle w^3, x^1 \rangle = 0.6*(1) + (-1.1)*(0) + (-0.4)*(1) = 0.2$ ，再利用公式(2)，

求輸出 $o(x^1) = \text{sign}(f^1) = \text{sign}(0.2) = 1$ ，此結果不等於 $y^1 = -1$ ，要用公式(4)來更新權重：

$$w^4 = w^3 + 0.1(-1 - (1))x^1$$

$$= w^3 - 0.2x^1 = \begin{pmatrix} 0.6 \\ -1.1 \\ -0.4 \end{pmatrix} - 0.2 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.6 \\ -1.1 \\ -0.4 \end{pmatrix} - \begin{pmatrix} 0.2 \\ 0 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 0.6-0.2 \\ -1.1-0 \\ -0.4-0.2 \end{pmatrix} = \begin{pmatrix} 0.4 \\ -1.1 \\ -0.6 \end{pmatrix}$$

累積錯誤 $E = E + 0.5 * ||y^1 - o(x^1)||^2 = 0 + 0.5 * ||-1 - (1)||^2 = 0 + 0.5 * ||-2||^2 = 0.5 * 4 = 2$ 。

5. 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 $f^2 = \langle w^4, x^2 \rangle = 0.4*(0) + (-1.1)*(-1) + (-0.6)*(-1) = 1.7$ ，再利用公式(2)，求輸出 $o(x^2) = \text{sign}(f^2) = \text{sign}(1.7) = 1$ ，此結果等於 $y^2 = 1$ ，權重不用更新，即 $w^5 = w^4$ 。
累積錯誤 $E = E + 0.5 * ||y^2 - o(x^2)||^2 = 2 + 0.5 * ||1 - 1||^2 = 2$ 。
6. 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 $f^3 = \langle w^5, x^3 \rangle = 0.4*(-1) + (-1.1)*(-0.5) + (-0.6)*(-1) = 0.75$ ，再利用公式(2)，求輸出 $o(x^3) = \text{sign}(f^3) = \text{sign}(0.75) = 1$ ，此結果等於 $y^3 = 1$ ，權重不用更新，即 $w^6 = w^5$ 。
累積錯誤 $E = E + 0.5 * ||y^3 - o(x^3)||^2 = 2 + 0.5 * ||1 - 1||^2 = 2$ 。
7. 累積錯誤 $E > 0$ ， $E = 0$ ， $k = 1$ ，跳到 Step 3，啟動新的訓練週期。
 $\langle w^6, x^1 \rangle = (0.4)*(1) + (-1.1)*(0) + (-0.6)*(1) = -0.2 < 0$ ，即 $\text{sign}(-0.2) = -1 = y^1 = -1$ ， $E = 0$ 。
 $\langle w^6, x^2 \rangle = (0.4)*(0) + (-1.1)*(-1) + (-0.6)*(-1) = 1.7 > 0$ ，即 $\text{sign}(1.7) = 1 = y^2 = 1$ ， $E = 0$ 。
 $\langle w^6, x^3 \rangle = (0.4)*(-1) + (-1.1)*(-0.5) + (-0.6)*(-1) = 0.75 > 0$ ，即 $\text{sign}(0.75) = 1 = y^3 = 1$ ， $E = 0$ 。
累積錯誤 $E = 0$ ，結束訓練，最後得到的權重為 $w^6 = (0.4, -1.1, -0.6)$ 。
8. 再來利用訓練得到的權重，來測試新的輸入命令 $x^4 = (1, 1, 1)$ ，帶入公式(1)，此時的 f 為 $f^4 = \langle w^6, x^4 \rangle = (0.4)*(1) + (-1.1)*(1) + (-0.6)*(1) = -1.3$ ，再利用公式(2)，求輸出 $o(x^4) = \text{sign}(f^4) = \text{sign}(-1.3) = -1$ ，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。

全國高級中等學校106學年度工業類科學生技藝競賽

檔案

工作桌編號 _____ 選手姓名 _____ 代表學校 _____

機器人訓練

輸入訓練資料對		
輸入命令x	輸出指令y	
1	0	1
0	-1	-1
-1	-0.5	-1

訓練參數

最大訓練次數

學習率

初始權重值

訓練後得到的權重值

測試輸入命令

新的命令

機器人向:

- (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)。

全國高級中等學校106學年度工業類科學生技藝競賽

檔案

工作桌編號 _____ 選手姓名 _____ 代表學校 _____

機器人訓練

輸入訓練資料對

輸入命令x	輸出指令y		
1	0	1	-1
0	-1	-1	1
-1	-0.5	-1	1

訓練參數

最大訓練次數

學習率

初始權重值

訓練後得到的權重值

測試輸入命令

新的命令

機器人向:

(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)。

全國高級中等學校106學年度工業類科學生技藝競賽

檔案

工作桌編號 _____ 選手姓名 _____ 代表學校 _____

機器人訓練

輸入訓練資料對

輸入命令x	輸出指令y		
1	0	1	-1
0	-1	-1	1
-1	-0.5	-1	1

訓練參數

最大訓練次數

學習率

初始權重值

訓練後得到的權重值

測試輸入命令

新的命令

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 $Y_t = a + bx$ ，式中 x 是自變數，是選定的任何 x 值； Y_t 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。

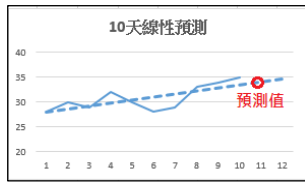


圖 5-1 以 10 天交易價的預測線

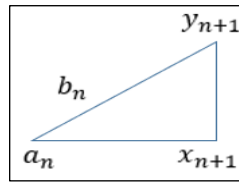


圖 5-2 預測次日的 y_{n+1}

日期系列:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
價格系列::	40.0	41.0	42.0	32.0	30.0	28.0	29.0	33.0	34.0	35.0	34.0	32.0	33.0	35.0	34.0
直線斜率b:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.9	-0.6	-0.2	0.4	0.6	0.5
價格預測:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	29.4	30.7	31.9	34.4	35.6

圖 5-3 一系列預測值計算的結果

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

■平均值：

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

■迴歸係數：

$$b = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$a = \bar{y} - b\bar{x}$$

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 (x_{n+1}) 的 y_{n+1} 之變動趨勢，亦即 $y_{n+1} = a_n + b_n x_{n+1}$ ，如圖 5-2 所示，其中 a_n 、 b_n 為某日之迴歸係數， x_{n+1} 為某日之次日， y_{n+1} 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：

日期系列:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
價格系列::	25.0	27.0	29.0	32.0	35.0	34.0	31.0	33.0	32.0	31.0	29.0	28.0	30.0	33.0	32.0

輸出格式：

直線斜率b:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.1	-0.3	-0.6	-0.4	-0.2
價格預測:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	34.4	32.1	29.6	28.4	29.1	

操作範例 1：

輸入：

日期系列:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
價格系列::	25.0	27.0	29.0	32.0	35.0	34.0	31.0	33.0	32.0	31.0	29.0	28.0	30.0	33.0	32.0

輸出：

直線斜率b:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.1	-0.3	-0.6	-0.4	-0.2
價格預測:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	34.4	32.1	29.6	28.4	29.1

操作範例 2：

輸入：

日期系列:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
價格系列::	40.0	41.0	42.0	32.0	30.0	28.0	29.0	33.0	34.0	35.0	34.0	32.0	33.0	35.0	34.0

輸出：

直線斜率b:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.9	-0.6	-0.2	0.4	0.6	0.5
價格預測:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	29.4	30.7	31.9	34.4	35.6

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 ($2 \leq n \leq 100$)
 - A. 對每一個輸入 n ，第一行輸出 $n!$ 的值。例如： $8! = 40320$ 。
 - B. 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 $2^{31}-1$ 。例如： $8! = 40320$ ，每個位數的總和為 $4+0+3+2+0 = 9$ 。
 - C. 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： $8! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 = 2 \cdot 3 \cdot 2 \cdot 2 \cdot 5 \cdot 3 \cdot 2 \cdot 7 \cdot 2 \cdot 2 \cdot 2 = 2^7 \cdot 3^2 \cdot 5 \cdot 7$ ，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 $8!$ 。
2. 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 ($2 \leq n \leq 100$)。請參考範例:
3. 範例輸入
8
12
20
100
4. 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。
5. 範例輸出
40320
9
11

479001600
27
19

2432902008176640000
54
36

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463
9761565182862536979208272237582511852109168640000000000000000000000000
648
239
6. 評分標準
 - A. 介面不拘，可以輸入測資並輸出結果即可。
 - B. 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。