

試題一：劃線乘法。(17分)

說明：乘法用畫斜平行線交叉的方式，可數算交叉點數，來算出兩數相乘的答案。被乘數由左至右畫斜平行線，斜線角度約 45° (誤差約 $\pm 5^\circ$)，乘數由下至上畫斜平行線，斜線角度約 135° (誤差約 $\pm 5^\circ$)，十位數和個位數應有適當間隔，直接觀察交叉點數可得相乘的答案，如圖 1-1 和 1-2 所示。請設計一程式能輸入被乘數及乘數，被乘數的範圍：1~50，乘數的範圍：1~50，能畫出交叉圖，被乘數用紅色畫線，乘數用藍色畫線。若遇 0 時，以虛線取代實線如圖 1-2 所示。虛線和虛線或虛線和實線交叉皆代表 0。

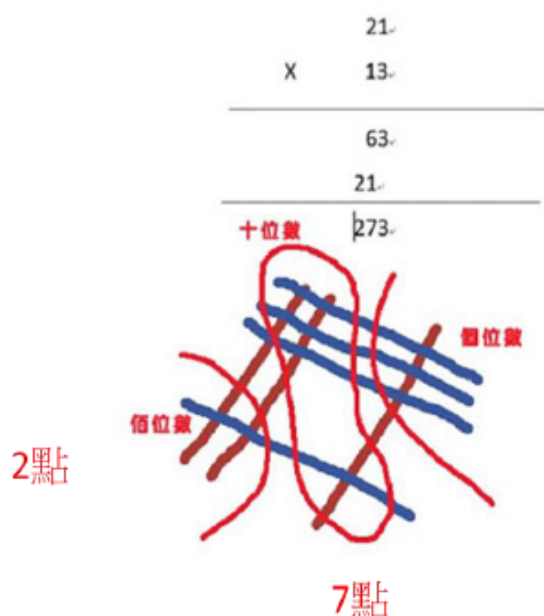


圖 1-1

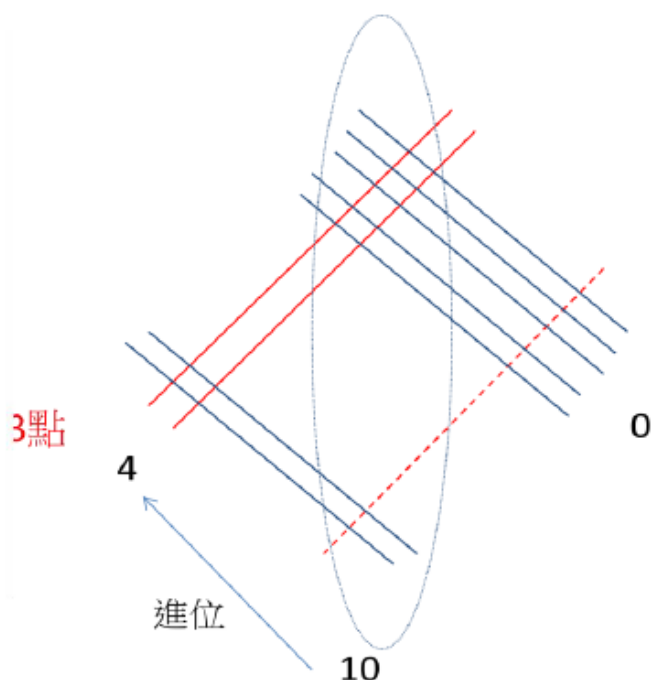


圖 1-2 (20X25=500)

圖形介面參考如圖 1-3 所示。

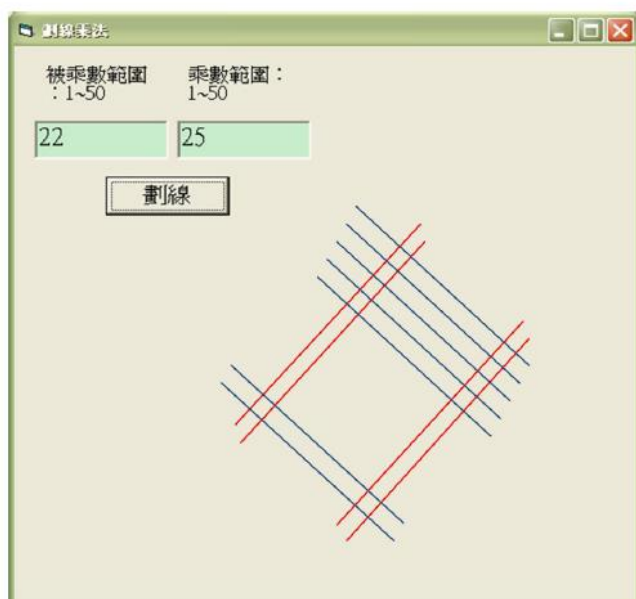
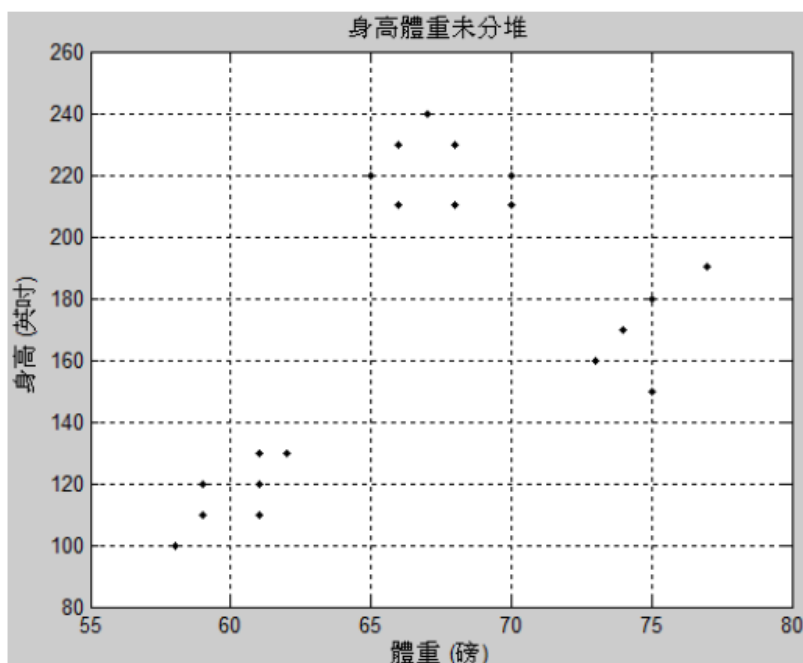


圖 1-3 圖形介面

試題二：分堆問題(16 分)

說明：分堆問題是把聚在一起的資料，當作一堆，如下圖所示，為 20 個人之身高和體重之分佈，其中，水平軸是體重，垂直軸是身高，由此圖，我們可以知道有三堆，每一堆的身高和體重都差不多，但是電腦要如何知道有三堆呢？



有一個分堆演算法如下：

- 輸入 N 個人(x_i)之身高 H_i 和體重 W_i 資料, $1 \leq i \leq N$ 。
- 因為每個人之身高 H_i 和體重 W_i ，其單位不一樣，所以，以下列式子，來得到正規化之身高 NH_i 和體重 NW_i 。

$NH_i = \frac{H_i - \mu_H}{\sigma_H} \quad (1)$	$NW_i = \frac{W_i - \mu_W}{\sigma_W} \quad (2)$
---	---

其中， $\mu_H, \mu_W, \sigma_H, \sigma_W$ ，分別定義如下：

$\mu_H = \frac{\sum_{i=1}^N H_i}{N} \quad (3)$	$\mu_W = \frac{\sum_{i=1}^N W_i}{N} \quad (4)$	$\sigma_H = \frac{\sum_{i=1}^N (H_i - \mu_H)^2}{N} \quad (5)$	$\sigma_W = \frac{\sum_{i=1}^N (W_i - \mu_W)^2}{N} \quad (6)$
--	--	---	---

- 初始分三堆 $S_j, 0 \leq j \leq 2$: 第一個人之身高和體重初始在第一堆 S_0 , 第二個人之身高和體重初始在第二堆 S_1 , 第三個人之身高和體重初始在第三堆 S_2 , 剩下的人，用亂數來初始分入這三堆中。

4. 執行以下步驟 200 次，若是分堆穩定，即每一堆都沒有異動了，表示分堆完成，就提早結束。
5. 在第 t 次中，計算第 j 堆 $S_j^{(t)}$ 中，所有人之平均身高 $u_j^{(t)}(H)$ 和平均體重 $u_j^{(t)}(W)$ ，其公式分別如下：

$u_j^{(t)}(H) = \frac{1}{ S_j^{(t)} } \sum_{x_i \in S_j^{(t)}} x_i(H_i) \quad (7)$	$u_j^{(t)}(W) = \frac{1}{ S_j^{(t)} } \sum_{x_i \in S_j^{(t)}} x_i(W_i) \quad (8)$
--	--

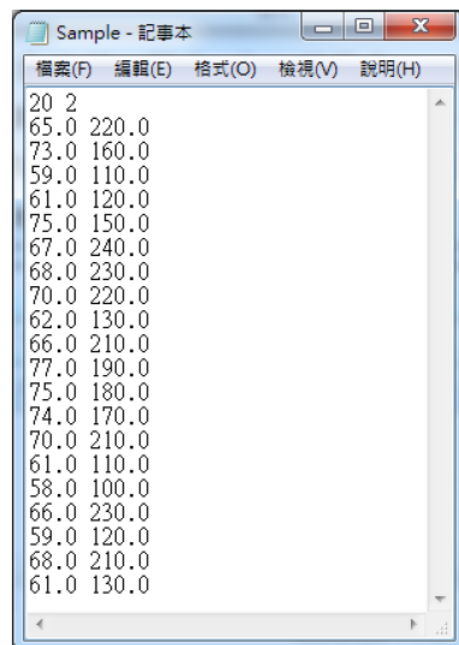
6. 其中， $x_i(H_i)$ 和 $x_i(W_i)$ 分別為第 x_i 位人之身高 H_i 和體重 W_i ， $x_i \in S_j^{(t)}$ 是第 x_i 位人屬於在第 t 次中，第 j 堆 $S_j^{(t)}$ 之人。
- 例如，假設有 d0, d1, d2 三個人聚成一堆，其體重和身高分別為 d0 = {64, 110}, d1 = {65, 160}, d2 = {72, 180}。則這一堆之平均體重和平均身高為 $\{(64+65+72)/3, (110+160+180)/3\} = \{67.0, 150.0\}$ 。
7. 依照這新的每堆之平均身高 $u_j^{(t)}(H)$ 和平均體重 $u_j^{(t)}(W)$ ，重新更動每一堆的人選，其做法是將第 x_i 位人之身高和體重與第 j 堆之平均身高 $u_j^{(t)}(H)$ 和平均體重 $u_j^{(t)}(W)$ 計算距離 $d_j(x_i)$ ，其公式如下：

$d_j(x_i) = \sqrt{\sum_{x_i \in S_j^{(t)}} \ x_i - \mu_j^{(t)}\ ^2} \quad (9)$	$d_{j^*}(x_i) = \arg \min_{0 \leq j \leq 2} d_j(x_i) \quad (10)$
--	--

其中， $\arg \min$ 表示從 j 堆距離中，找出第 x_i 人距離哪一堆最近， j^* 表示，第 x_i 人要重新分配到第 j 堆 $S_j^{(t)}$ 中。例如，假設有一 d0 人之身高和體重為 {140, 68}，假設有三堆 c0, c1, c2 其平均身高和平均體重分別為 c0 = {120.0, 66.0}, c1 = {160.0, 69.0}, 和 c2 = {130.0, 70.0}，在此我們用未正規化資料來說明，d0 和 c0 之距離為 $\sqrt{(68 - 66.0)^2 + (140 - 120.0)^2} = 20.10$ ，而 d0 和 c1 之距離為 $\sqrt{(68 - 69.0)^2 + (140 - 160.0)^2} = 20.22$ ，d0 和 c2 之距離為 $\sqrt{(68 - 70.0)^2 + (140 - 130.0)^2} = 10.20$ 。所以，d0 最靠近 c2，也就是 d0 要重新分配到第 2 堆中，sqrt 表示開根號。

程式功能：請利用上述演算法，寫一個程式，能完成以下功能要求：

- (1) 能讓使用者挑選文字檔，此文字檔之範例 Sample.txt 如下：



其中，第一列之 20，表示有 20 個人，第一列之 2 表示每個人有體重和身高 2 個資料，第二列起是這 20 個人之體重和身高。

(2) 能讓正確分類。程式執行範例：

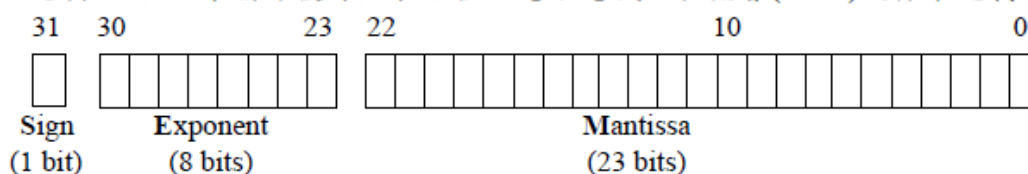
第0堆			第1堆			第2堆		
筆數	體重	身高	筆數	體重	身高	筆數	體重	身高
1	73	160	0	65	220	2	59	110
4	75	150	5	67	240	3	61	120
10	77	190	6	68	230	8	62	130
11	75	180	7	70	220	14	61	110
12	74	170	9	66	210	15	58	100
			13	70	210	17	59	120
			16	66	230	19	61	130
			18	68	210			

從上圖左邊開始，按 1 載入資料，可以讓使用者選擇輸入之文字檔，此例是選擇 Sample.txt，按 2 執行分堆，會依照您所寫之程式，進行依照體重與身高來進行分堆動作，分堆結果，顯示分堆之結果，第 0 筆分到第 1 堆，在第 1 堆中確實有顯示此筆之體重與身高資料，第 1 筆分到第 0 堆，在第 0 堆中確實有顯示此筆之體重與身高資料，第 2 筆分到第 2 堆，在第 2 堆中確實有顯示此筆之體重與身高資料，其他筆，依此列推，最後分堆結果，顯示在上圖右邊。

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 三：單精確度浮點二進制值轉換為十進制實數值系統(17 分)

說明：一個二進制 32 位元單精確度浮點表示值，電子電機工程協會(IEEE)的標準定義如下：



最左邊的 bit 31 為一個符號位元 S (Sign)，S 為 0 表示正值，而 S 為 1 則為負值；中間 bits 30~23 為八個位元之指數值 E (Exponent)，採用超 127 (Excess-127)格式，即原有指數值再外加 127，可將 2 的指數次方 -127 至 +128 改以 2 的指數次方 0 至 255 來表示；最右邊 bits 22~0 為 23 個位元之定點數值 M (Mantissa)，此正規化(Normalization)將第 22 位元左邊隱藏了小數點及小數點左邊一個 1，例如 1.011101...，而 23 個位元只記錄 011101...。

例題：一個二進制 32 位元單精確度浮點表示值如下所示，將它轉換為十進制實數值 R。

1 10000110 010000111100000000000000

轉換方法分為四個步驟敘述如下：

Step1：首先得知最左邊 bit 31 之符號位元為 $S = 1$ ，可判斷此十進制實數值為負值；

Step2：其次得知中間 bits 30~23 之八個位元超 127 指數值為 $E = 10000110_2 = 134_{10}$ ，將此超 127 指數值 E 還原回來的 $E = 134 - 127 = 7$ ，即 2 的指數次方 7，也就是 2^7 ；

Step3：接著得知最右邊 bits 22~0 之 23 個位元定點數值為 $M = 01000011110000000000000_2$ ，將此正規化 M 值做反正規化(還原隱藏的小數點及小數點左邊一個 1)，則還原回來的 $M = 1.0100001111000000000000_2$ ；

Step4：最後還原回來二進制浮點表示為 $-1.0100001111000000000000_2 \times 2^7 = -10100001.1110000000\ 00000_2$ ，再將此值轉換為十進制實數值 $R = -161.875$ 。

請參考以上例題與轉換方法，設計如下圖所示之系統，當滑鼠點一下 **Random** 鍵，該系統在 IEEE Excess-127 右方格子內自動產生對應 bit31、bits 30~23 及 bits 22~0 等二進制 32 位元單精確度浮點表示值，且自動清除 Real number 右方格子內容；另可任意輸入或更改二進制 32 位元單精確度浮點表示值，但仍限制維持在 32 位元。當滑鼠點一下 **Convert** 鍵，則將 IEEE Excess-127 右方二進制 32 位元單精確度浮點表示值轉換為等效的正或負十進制實數值，並顯示在 Real number 的右方格子，小數點值最多取十位。上述可重複操作，直至以滑鼠點一下 Exit 鍵，則離開此系統。

IEEE 超 127 單精確度浮點二進制值轉換為十進制實數值系統

IEEE Excess-127:

Real number:

Random **Convert** **Exit**

範例

輸入格式：當滑鼠點一下 **Random** 鍵或人工輸入，則在 IEEE Excess-127 右方格子內自動或人工產生對應正好 32 位元單精確度浮點二進值，且自動清除 Real number 右方格子內容。

輸出格式：當滑鼠點一下 **Convert** 鍵，則將 IEEE Excess-127 之 32 位元單精確度浮點二進值轉換為等效的正或負十進制實數值，顯示在 Real number 的右方格子內，小數點值最多取十位。

IEEE 超 127 單精確度浮點二進制值轉換為十進制實數值系統

IEEE Excess-127:

Real number:

Random **Convert** **Exit**

試題 四：資料序列之相似度計算程式(17 分)

說明：一、對於處理數字資料序列，可用於時間序列的處理，例如：多日的交易結果的成交值，也可用於聲頻的特徵值的處理，例如：聲頻的共振峰值。對已有的多筆數字的資料序列，我們可將其視為樣本資料，透過某種方法，可將這些樣本資料建立出一個樣本模型。若有一時間序列的資料，或聲頻的共振峰資料，我們即可將此資料與樣本模型進行比對，之後，可得一值，稱之為相似度，代表此資料與樣本模型接近的程度，相似度的值越大，接近的程度越高。

二、今有三個數值序列可用以表示樣本模型，詳見圖 7 下方 3 個數值序列，其中數值序列 2 的相似度為 1 (100%相似)，其曲線如圖 1 中間粗線所示；在圖 7 中，對於數值序列 2，以 $x=7$ 為例，數值為 19 時，則其相似度為 1。為了簡化解法，採用線性遞減的方式，數值 ≥ 22 時，則其相似度才降為 0；數值 ≤ 17 時，則其相似度也降為 0。其相似度三角圖形如圖 7 上方所示，隨著數值的大小，以直線方程式表示相似度的增減，且三角形的兩邊不一定是對稱。同理，在數值序列 2 中，再以 $x=15$ 為例，數值為 13 時，則其相似度為 1，數值 ≥ 16 或 ≤ 10 時，其相似度均降至 0。

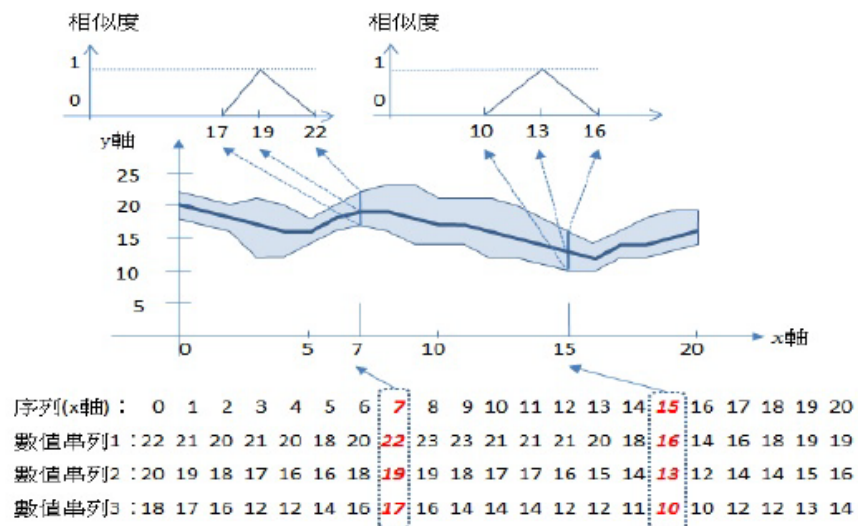


圖 7 樣本模型

三、由圖 7 所示，依三個數值序列，我們得到的相似度模型圖，如圖 7 中間灰階區域所示，中間粗線代表相似度為 1 的連線，上下兩邊的細線代表相似度降為 0 的連線。一旦獲得此相似度模型，我們即可用來計算一個未知的資料序列，以求得其相似度。

四、對於未知的資料序列(見圖 8 下方之未知序列)，在圖 8 中間以粗虛線段表示。其每一個資料都與樣本模型對應的資料比對，如同圖 7 之 $x=7$ 或 $x=15$ 的資料般，求其個別的相似度，最後，再求此 21 個相似度的平均相似度，即代表此未知的資料序列與樣本模型相近的程度。

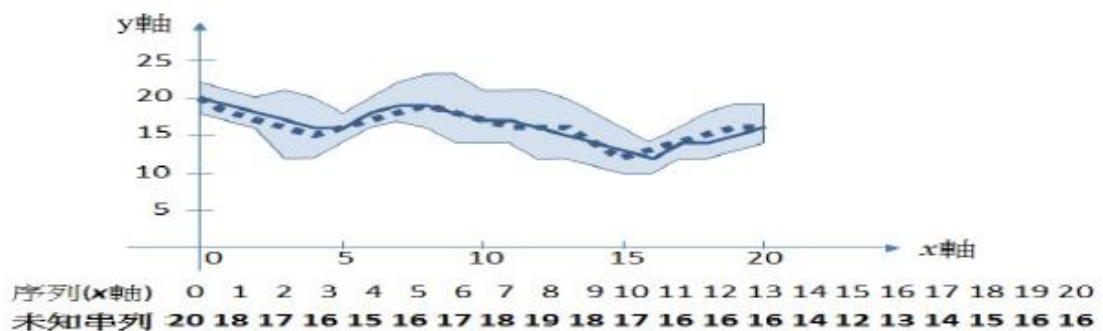


圖 8 未知的資料序列與相似度模型之比對

五、設計步驟：

1. 依據 3 個資料串列建立相似度模型(如圖 7 中間灰階區域)。
2. 輸入未知的資料串列(如圖 8 之粗虛線段者)，進行未知的資料串列與相似度模型之比對。
3. 輸出計算的結果。

輸入及輸出格式：

輸入格式(模型)：

22	21	20	21	20	18	20	22	23	23	21	21	21	20	18	16	14	16	18	19	19
20	19	18	17	16	16	18	19	19	18	17	17	16	15	14	13	12	14	14	15	16
18	17	16	12	12	14	16	17	16	14	14	14	12	12	11	10	10	12	12	13	14

輸入格式(資料串列 1)：

15	15	15	15	15	15	15	15	15	15	15	15	18	18	18	18	18	18	18	18	18
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

輸出格式： 0.226984

或

輸入格式(資料串列 2)：

20	18	17	16	15	16	17	18	19	18	17	16	16	16	14	12	13	14	15	16	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

輸出格式： 0.794445

操作畫面：

a. 主選單

請選擇操作項目：
(1)輸入模型資料：
(2)計算平均相似度：
(3)顯示各資料相似度：
請選擇：1

b. 選項操作：輸入模型資料

請選擇操作項目：
(1)輸入模型資料：
(2)計算平均相似度：
(3)顯示各資料相似度：
請選擇：1
輸入模型資料，總筆數為：21
序列< x軸>： 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
數值串列<上限>： 22 21 20 21 20 18 20 22 23 23 21 21 21 20 18 16 14 16 18 19 19
數值串列<中心>： 20 19 18 17 16 16 18 19 19 18 17 17 16 15 14 13 12 14 14 15 16
數值串列<下限>： 18 17 16 12 12 14 16 17 16 14 14 14 12 12 11 10 10 12 12 13 14
繼續：請按1，結束：請按0：1

c. 選項操作：計算平均相似度

請選擇操作項目：
(1)輸入模型資料：
(2)計算平均相似度：
(3)顯示各資料相似度：
請選擇：2
請輸入「資料串列」檔名：4.txt
已開啟「資料串列」檔名：4.txt
平均相似度為 0.794445
繼續：請按1，結束：請按0：1

試題 五：2 維卷積(2D Convolution) (17 分)

說明：在求線性和時間不變系統的輸出 $O[m, n]$ 時，常將輸入的 2 維信號 $I[m, n]$ and 核 $K[m, n]$ 做 2 維卷積，這可以表示如下式子：

$$O[m, n] = I[m, n] \otimes K[m, n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} I[i, j] \times K[m-i, n-j] \quad (1)$$

核 $K[m, n]$ 的中心是 $K[0, 0]$ ，假如，核的大小是 3，此核的索引是 -1, 0, 1，所以， $K[0, 0]$ 是在核的中間。假設有一核如圖 9 所示，此核大小是 3×3 ，即 $m=3, n=3$ ，核中之 9 個值為 a, b, c, \dots, i ，核的原點是 $(0, 0)$ ，是落在核的中央。假設要計算輸出 $m=1, n=1$ 時，帶入公式(1)，可以用圖 10 來表示，注意，在做 2 維卷積時，核(kernel)需要翻動(flipped)，再跟輸入(input)重疊部分相乘。例如，如圖 10 所示，輸入 $I[0, 0]$ 是和核中的 i 相乘，輸入 $I[2, 2]$ 是和核中的 a 相乘。

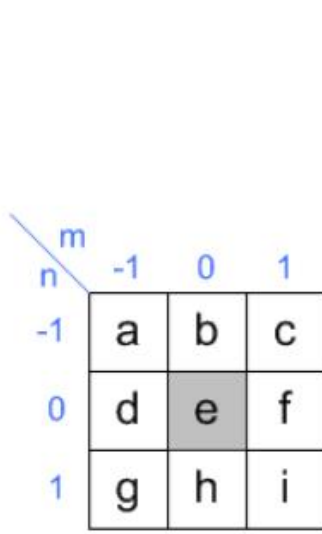


圖 9

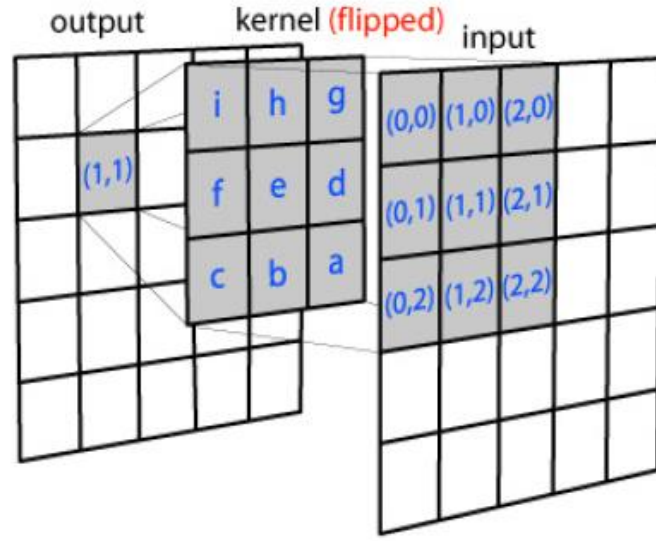


圖 10

另外，為了判斷輸入和輸出的相似度，常用的方法有以下三種方法：

1. 平均平方誤差 (Mean Square Error, MSE):

$$MSE = \frac{1}{WH} \sum_{y=1}^H \sum_{x=1}^W [I[x, y] - O[x, y]]^2 \quad (2)$$

2. 平均絕對誤差 (Mean Absolute Error, MAE):

$$MAE = \frac{1}{WH} \sum_{y=1}^H \sum_{x=1}^W |I[x, y] - O[x, y]| \quad (3)$$

3. 峰值訊號雜訊比 (Peak Signal to Noise Ratio):

$$PSNR = 10 \log_{10} \frac{255 * 255}{MSE} \quad (4)$$

上述 W 和 H 分別表示為寬度和高度， $|\cdot|$ 表示絕對值。

請利用上述說明，寫一個程式，能完成以下功能要求：

- (1) 能讓使用者輸入 7x7 資料，這些資料要大於等於 0，小於等於 255。
- (2) 能讓使用者輸入 3x3 核。
- (3) 能讓使用者按<運算>執行，得到輸出 0。
- (4) 將輸出 0 顯示出來。
- (5) 計算輸出 0 和輸入 I 之相似度 (MSE , MAE , $PSNR$)，並顯示出來。

程式執行範例：

全國高級中等學校104學年度工業類科學生技藝競賽

工作桌編號 _____ 選手姓名 _____ 代表學校 _____

1.輸入I

0	1	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	1	2	3	0
3	0	0	4	5	6	0
4	0	0	7	8	9	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

2.核K設定

-1	0	1
-1	-2	-1
0	0	0
1	2	1

3.運算

MSE = _____

MAE = _____

PSNR = _____

4.輸出O

0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

上圖從左邊開始，第 1 讓使用者輸入 7x7 輸入 I 資料，第 2 讓使用者設定 3x3 核 K。

全國高級中等學校104學年度工業類科學生技藝競賽

工作桌編號 _____ 選手姓名 _____ 代表學校 _____

1.輸入I

0	1	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	1	2	3	0
3	0	0	4	5	6	0
4	0	0	7	8	9	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

2.核K設定

-1	0	1
-1	-2	-1
0	0	0
1	2	1

3.運算

MSE = 119.775510204082

MAE = 6.30612244897959

PSNR = 27.3471233138285

4.輸出O

0	0	0	0	0	0	0
1	0	-1	-4	-8	-8	-3
2	0	-4	-13	-20	-17	-6
3	0	-6	-18	-24	-18	-6
4	0	4	13	20	17	6
5	0	7	22	32	26	9
6	0	0	0	0	0	0

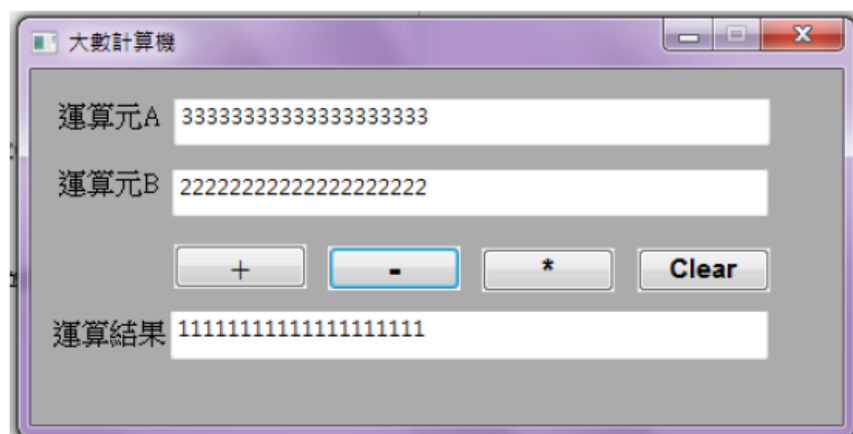
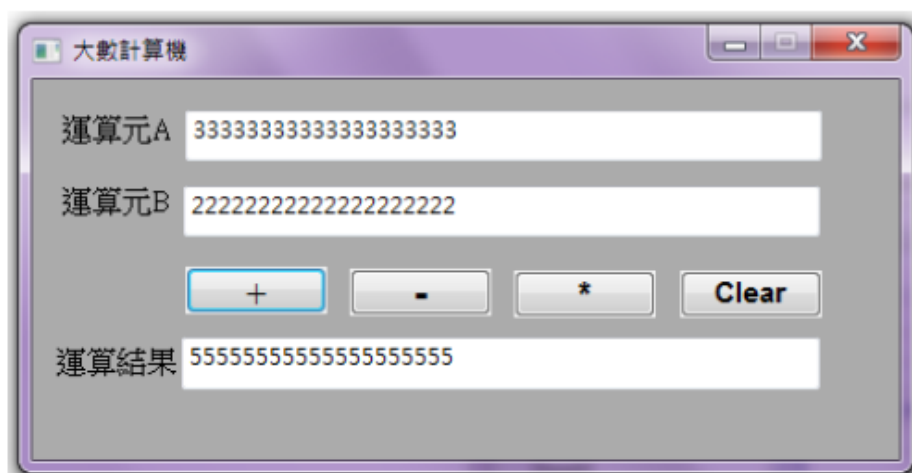
上圖是，第 3 讓使用者按<運算>執行，得到輸出 0。第 4 將輸出 0 顯示出來。同時，顯示 MSE , MAE , $PSNR$ 出來。

試題二：大數計算機，可以計算加法、減法與乘法的結果。

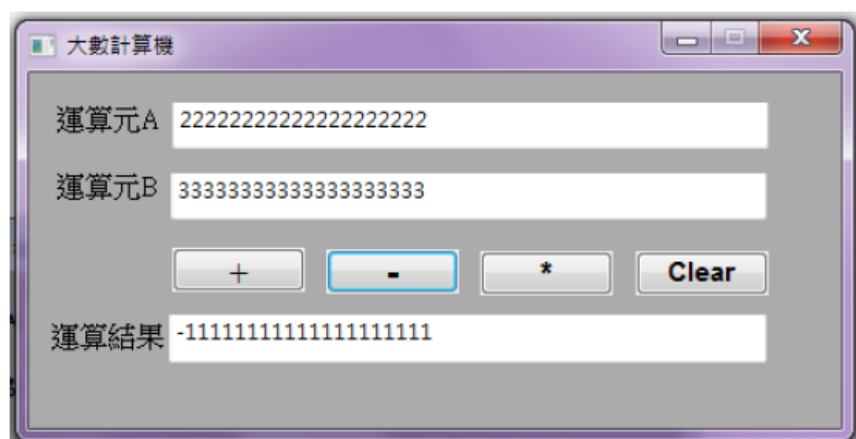
說明：

- 1 輸入的運算元與輸出的運算結果可能會超過整數型態的定義範圍，也就是會有溢位(overflow)的可能性，你的程式要可以處理超過整數型態的定義範圍的大數。
- 2 以 + 代表加法，以 - 代表減法，以 * 代表乘法
- 3 輸入說明：window 介面，可以輸入兩個正整數，最長為 20 個位數，可以顯示加法、減法與乘法的結果。
- 4 輸出說明：兩個正整數的運算結果，總長度不超過 40 個位數
- 5 參考範例：

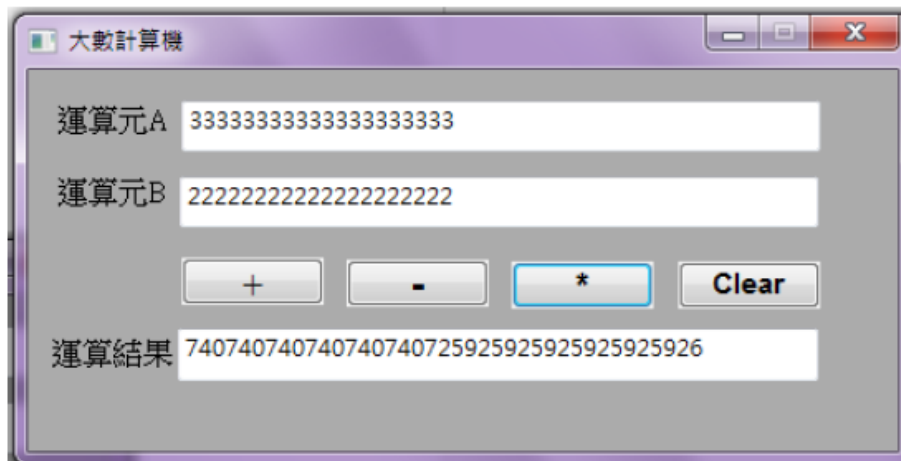
5.1 $333333333333333333 + 222222222222222222$



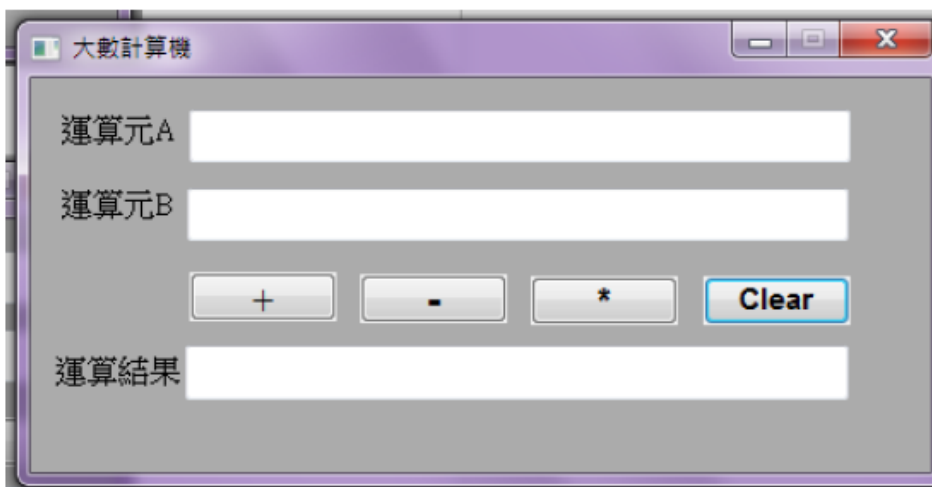
5.3 $222222222222222222 - 333333333333333333$



5.4 33333333333333333333 * 22222222222222222222



5.5 Clear 可以清除三個文字框



6 解題說明：

- 6.1 第一步驟可以宣告兩個線性陣列(linear array)分別儲存這兩個運算元，每個陣列的元素用來儲存一個十進制的位元，例如宣告一個陣列 `int a[3]` 來儲存 123，其中 `a[2] = 1`, `a[1] = 2`, `a[0] = 3`。再宣告另一個陣列 `int b[3]` 來儲存 789，其中 `b[2] = 7`, `b[1] = 8`, `b[0] = 9`。
- 6.2 第二步驟將兩個陣列的每個元素分別加起來，放到另外一個陣列 `int c[3]`，則 `c[2] = 8`, `c[1] = 10`, `c[0] = 12`。
- 6.3 第三步驟將陣列 `c` 的每個元素進行「逢十進位」的正規化，則 `c[0] = 2`, `c[1] = 1`, `c[2] = 9`。
- 6.4 第四步驟從索引高到低輸入陣列 `c` 的數值為 912。
- 6.5 減法運算要注意兩個運算元的大小。

7 評分標準

- 7.1 加法運算功能正確 5 分
- 7.2 減法運算功能正確 5 分
- 7.3 乘法運算功能正確 10 分
- 7.4 Clear 功能正確 5 分