14 葉[illegible]木 11/27

全國高級中等學校102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。

試題一：局部編碼(17分)

說明：局部編碼有許多應用，局部編碼(LC)是在一個資料區域內，利用相鄰鄰居和不同權重，來進行編碼。局部編碼可以利用下述公式來表示:

$$LC_{I,R}(x_c,y_c)=\sum_{i=0}^{I-1}T(d_i-d_c)\times 2^i \quad (1)$$

$$T(x)=\begin{cases}1 & x\geq 0\\ 0 & x<0\end{cases} \quad (2)$$

其中，d_c是在區域(R)中的中心點(x_c, y_c)之資料，d_i是區域中心點之 I 個鄰居資料點，d_i-d_c表示鄰居資料點與中心點之差，2^i是鄰居資料點之對應權重，$T(x)$是一閥值函數，當 x 大於等於0時，$T(x)$=1，當 x 小於0時，$T(x)$=0。

程式功能：請利用上述公式(1)和公式(2)，寫一個程式，能完成以下功能之要求:

(1)能讓使用者輸入6x6資料，這些資料要大於等於0，不可以小於0。

(2)能讓使用者輸入3x3權重，這些權重是2的次方。

(3)能讓使用者輸入要編碼之3x3區域的左上角座標。

(4)程式可以計算和顯示要編碼之3x3區域的編碼結果。

程式執行範例：

葉榮木 11/23

範例說明:從上圖左邊開始，第 1 步讓使用者輸入 6x6 資料，第 2 步讓使用者輸入 3x3 權重，第 3 步讓使用者輸入要編碼之 3x3 區域的左上角座標，座標請參考輸入資料之座標(0~5, 0~5)，第 4 步按計算執行，第 5 步顯示編碼結果。

上述範例，權重遮罩設定為$\begin{bmatrix} 1 & 2 & 4 \\ 8 & 0 & 16 \\ 32 & 64 & 128 \end{bmatrix}$，輸入要編碼之 3x3 區域的左上角座標為(1, 1)，也就是要對輸入區域資料$\begin{bmatrix} 55 & 35 & 28 \\ 52 & 43 & 38 \\ 26 & 65 & 46 \end{bmatrix}$進行編碼。再利用公式(2)運算後，其結果為$\begin{bmatrix} 1 & 0 & 0 \\ 1 & & 0 \\ 0 & 1 & 1 \end{bmatrix}$。再利用公式(1)運算後，其編碼結果為 201，如上圖所示。

<u>若妳（你）的程式都完成上述功能和要求，才可以上傳要求檢查功能。</u>

試題二：隨機實數值轉換為等效二進制值(16 分)

1. 設計一個程式系統能將隨機產生 0.0 ~ 9999.999999 之間的一個十進制實數值(Real value)轉換為等效二進制值(Binary value)。此實數值或二進制值其小數點左方的數值稱為整數部分(Integral part)，而小數點右方的數值稱為分數部分(Fraction part)。

2. 實數值的整數部分轉換為二進制值的方法:將其**整數部分 ÷2 取其餘數**插入二進制值左方，而其商值再÷2 **取**其餘數插入二進制值左方，依此類推，直至商值為 0；最後可得到實數值的整數部分之等效二進制值。

3. 實數值的分數部分轉換為二進制值的方法:將其**分數部分 ×2 取其整數**插入二進制值右方，而其新分數值再 ×2 取其整數插入二進制值右方，依此類推，直至新分數值為 0 或二進制值已具有 10 位數；最後可得到實數值的分數部分之等效二進制值。

4. 組合第 2 點與第 3 點轉換結果，即可得到十進制實數值轉換為等效二進制值。如果二進制值其小數點最右方存在無效 0，則必須將其移除，並修正此等效二進制值。

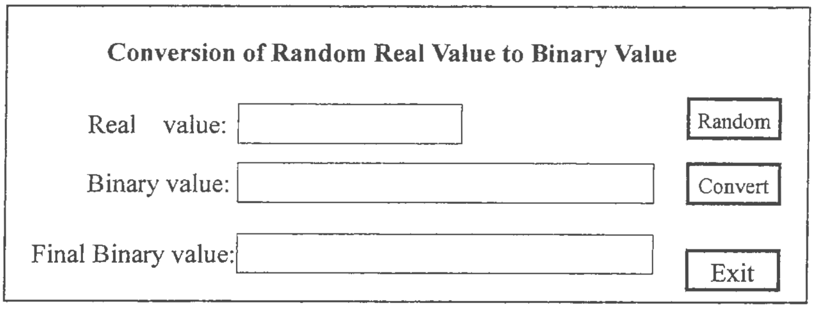
例題 1：十進制實數值 74.3312 轉換為等效二進制值 1001010.0101010011。

例題 2：十進制實數值 19.561 轉換為等效二進制值 10011.1000111110，經修正為 10011.100011111。

5. 設計此系統如下圖所示，每當滑鼠點一下 Random 鍵，則**隨機產**生 0.0 ~ 9999.999999 之間的一個十進制實數值顯示在 Real value 的右方,且此值可以人工修改(但限制在上述範圍內)。每當滑鼠點一下 Convert 鍵，則自動將 Real value 的十進制實數值轉換為等效二進制值，並顯示在 Binary value 的右方；如果二進制值其小數點最右方存在無效 0，則將其移除，且將修正後的二進制值顯示於 Final Binary value 的右方;如不存在無效 0,則 Final Binary value 的右方顯示值等同於原 Binary value。**上述可重複操作**，直至滑鼠點一下 Exit 鍵，則自動離

葉榮木 11/6

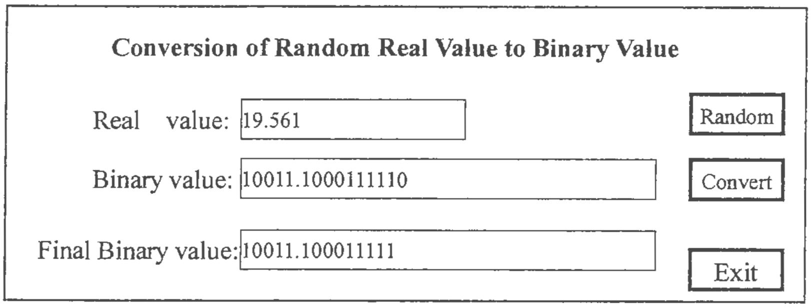
開此系統。



輸入格式：在 Real value 位置可隨機產生或人工輸入 0.0 ~ 9999.999999 之間任意一個的十進制實數值。

輸出格式：在 Binary value 及 Final Binary value 位置顯示此實數值的等效二進制值。

範例1： 在 Real value 位置隨機產生或人工輸入 19.561，滑鼠點一下 Convert 鍵，Binary value 及 Final Binary value 位置顯示如下：



試題三：在地圖中搜尋最低成本的路徑。(17分)

說明：1. 如圖 3-1 所示為一交通示意圖，圓圈節點代表城市，並以數字表示其順序；連接線表示城市間的行進方向，連接線的數字代表成本(成本=距離*時速)。某君規劃假期將由出發地到目的地(如示意圖，由節點①到節點⑦)，假設城市間的各路段之距離及時速可事先查詢完成，則某君在出發前即可將各路段之距離及時速換算為成本(如示意圖)，據以搜尋最低成本的路徑，並處理得知其所經各城市節點的先後次序，及其路徑的總和最低成本值。請設計一程式完成之。**範例一：**

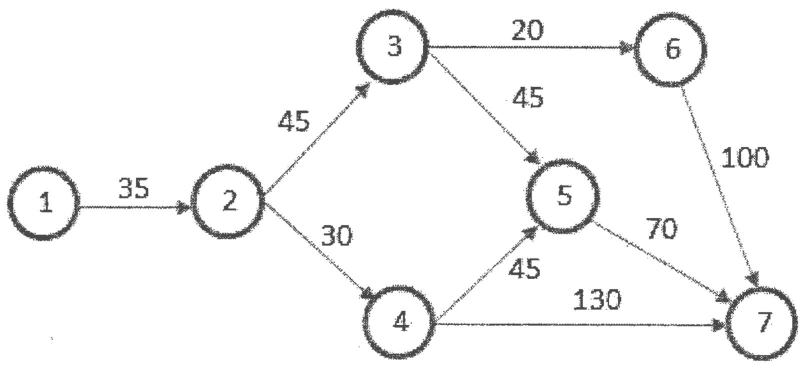


圖 3-1

2. 當城市間連接線的成本值改變，仍然可搜尋最低成本的路徑，並獲得其所經各城市節點的先後次序，及其路徑的總和最低成本值。

第14、17 1/2

輸入格式：1. 依序以「**城市編號　城市編號　城市間連接線的成本數字**」表示，例如：由節點①到節點②的表示為「1　2　35」。餘此類推，將所有城市及城市間連接線的數字設定完成，並存成輸入檔。

2. 輸入檔內容可隨時更改，格式如圖 3-2 所示

```
1 2 35
2 3 45
2 4 30
3 5 45
3 6 20
4 5 45
4 7 130
5 7 70
6 7 100
```

圖 3-2

```
最低成本值總和:180
路徑次序:  1  2  4  5  7
連線數值:  0 35 30 45 70
```

圖 3-3

3. 可隨時更改輸入檔的成本值，再搜尋最低成本的路徑次序及其最低成本值。

輸出格式：印出最低成本值總和、其所經各城市節點的先後次序及其路徑值，如圖 3-3 所示：

範例二：1. 節點⑥到節點⑦的值改為 70，如圖 3-4 所示：

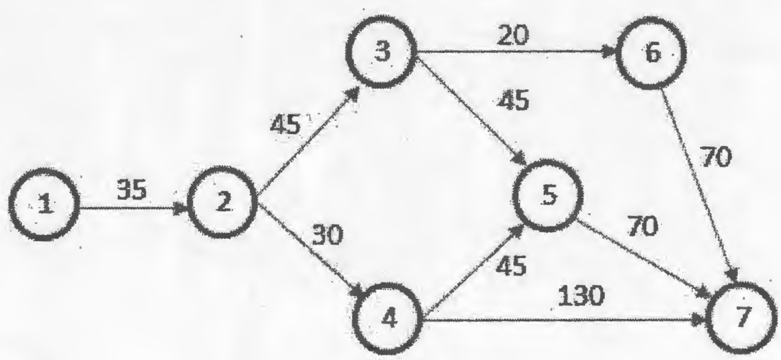


圖3-4

```
1 2 35
2 3 45
2 4 30
3 5 45
3 6 20
4 5 45
4 7 130
5 7 70
6 7 70
```

輸入格式：輸入檔更改內容，如上所示

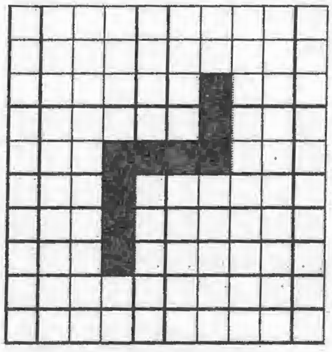
輸出格式：印出最低成本值總和、其所經各城市節點的先後次序及其路徑值，如下所示：

```
最低成本值總和:170
路徑次序:  1  2  3  6  7
連線數值:  0 35 45 20 70
```

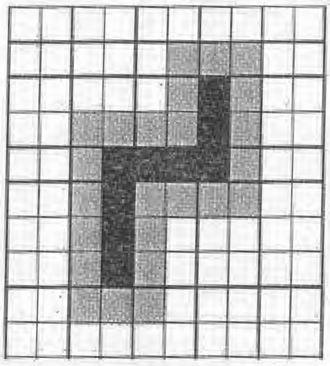
試題四：膨脹處理程式(17分)

說明：影像處理過程中常需進行膨脹(Dilation)，將影像向8方擴展(某像素的八個近鄰)，可用於去除雜點及彌補缺口，如下圖所示：

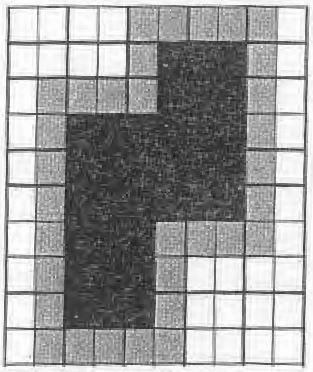
葉榮木 11/2



原圖

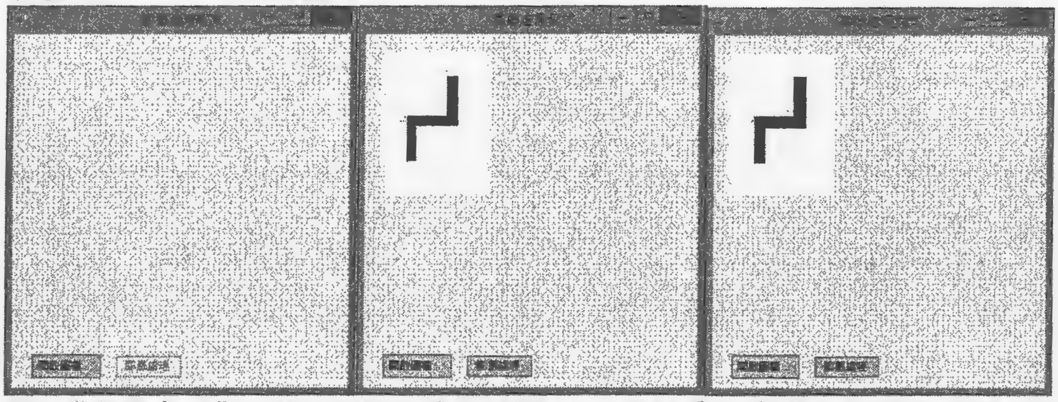


進行 1 次膨脹

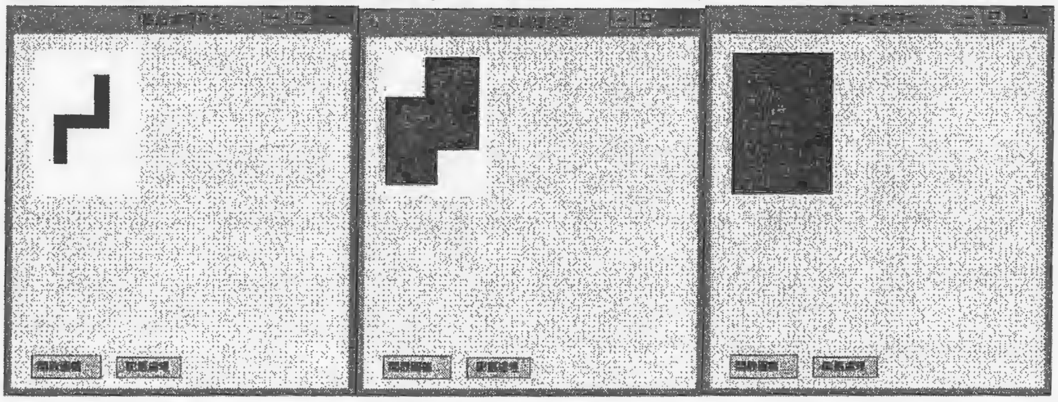


進行 2 次膨脹

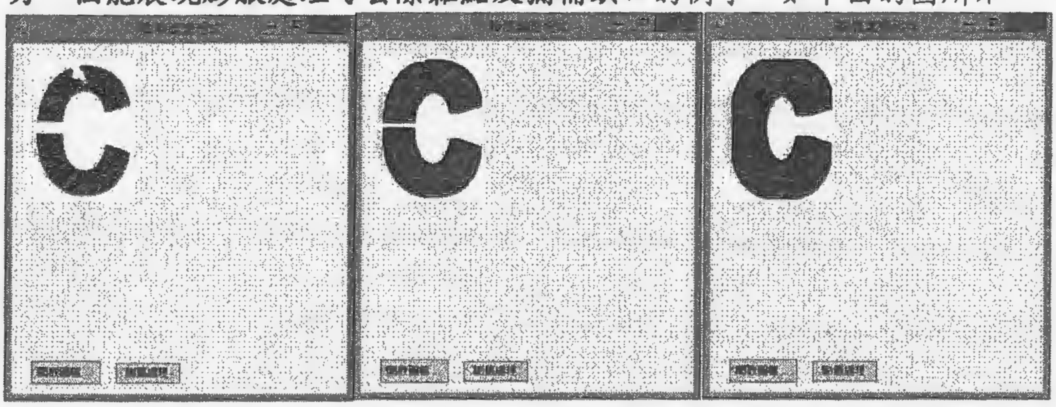
請寫隻程式能讀入圖檔，再按 “膨脹處理” 按鈕來進行圖案膨脹處理結果顯示在畫面上，如下面的圖所示：



按 “膨脹處理” 按鈕可進行多次膨脹，一直到最大膨脹，如下面的圖所示：



另一個能展現膨脹處理可去除雜點及彌補缺口的例子，如下面的圖所示：



葉榮木

試題五：印度小朋友的心算(16分)

說明:台灣小朋友會背９９乘法，印度的乘法表是從 1X1 背到 19X19，但 11X11 到 19X19 是用心算算出！印度的兩位數乘法是用心算算出，**但此心算必需被乘數跟乘數的十位數皆相同**。例子 1: 13X14＝？

第一步：先把被乘數（13）跟乘數的個位數（4）加起來，13＋4＝17

第二步:然後把第一步的答案乘以１０ （**十位數皆**是 1 也就是說後面加個 0 為 170）

第三步：再把被乘數的個位數（３）乘以乘數的個位數（4），3Ｘ4＝12

第四步：170＋12＝182

就這樣，用心算就可以很快地算出 11X11 到 19X19 了喔！

同理，例子 2: 26×27=?

第一步：先把被乘數（26）跟乘數的個位數（7）加起來，26＋7＝33

第二步：把第一步的答案乘以 20 （**十位數皆**是 2，乘 2 後，尾數再加個 0 為 660）

第三步：再把被乘數的個位數（6）乘以乘數的個位數（7），6Ｘ7＝42

第四步：660＋42＝702

就這樣，用心算就可以很快地算出 21Ｘ21 到 29Ｘ29 了喔！餘此可類推至更高位數！

小明想學印度的心算算法，請你(妳)設計一程式**能隨機產生被乘數 m 及乘數 n(被乘數和乘數的十位數需相同)，11≦m, n≦19 及 21≦m, n≦29** 如圖 5-1 讓小明心算作答，若小明**答正確，程式回饋"very good"**； 若小明**心算錯誤程式回饋"is wrong"並將正確的心算四個過程列出**如圖 5-2。**按清除需能出新題繼續作答。**

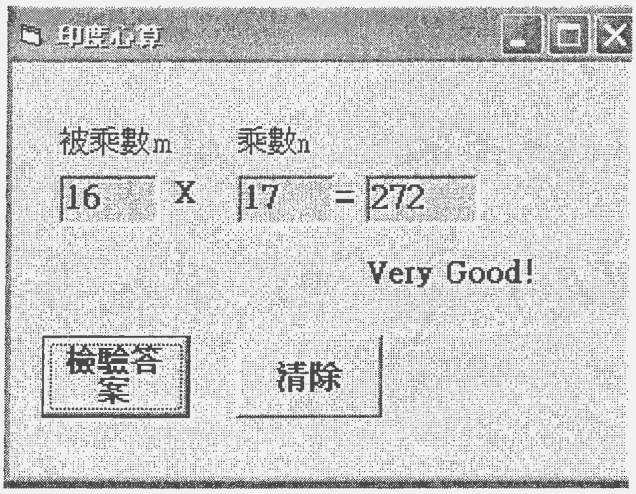


圖 5-1

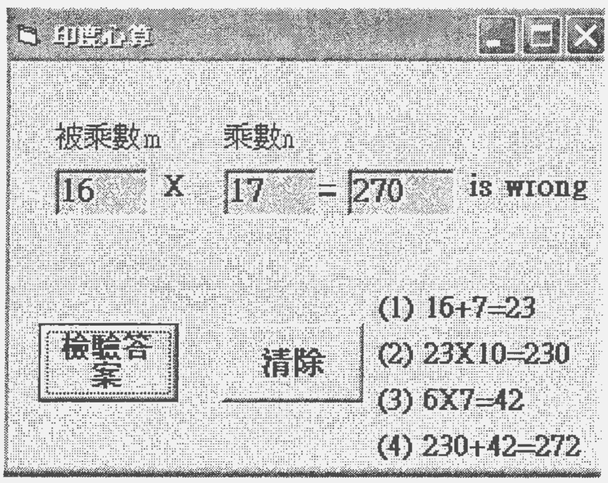


圖 5-2

試題六：多矩陣相乘運算次數(17 分)

說明：1. 兩個矩陣乘法運算必須滿足 A 矩陣的行數(2)等於 B 矩陣的列數(2)才可以相乘。**例如** A_{3X2} X B_{2X3}

$A=\begin{bmatrix}1 & 4\\ 2 & 3\\ 3 & 2\end{bmatrix}$ $B=\begin{bmatrix}1 & 2 & 3\\ 2 & 0 & -1\end{bmatrix}$	$A\times B=\begin{bmatrix}1\times1+4\times2 & 1\times2+4\times0 & 1\times3+4\times(-1)\\ 2\times1+3\times2 & 2\times2+3\times0 & 2\times3+3\times(-1)\\ 3\times1+2\times2 & 3\times2+2\times0 & 3\times3+2\times(-1)\end{bmatrix}=\begin{bmatrix}9 & 2 & -1\\ 8 & 4 & 3\\ 7 & 6 & 7\end{bmatrix}$ **A_{3X2} X B_{2X3} 的乘法運算次數為 3x2x3=18 次；**

葉榮木 11/23

2. 給定 n 個矩陣 $M_1, M_2, M_3, M_4, \cdots, M_n$，執行矩陣相乘運算，**其中，每一矩陣之行數**(column)**與列數**(row)**均不大於** 100，**且** $3 \leq n \leq 10$。請你(妳)設計一程式**能輸入** n **個可相乘矩陣大小**，找到這些矩陣相乘時($M_1 \times M_2 \times M_3 \times M_4 \times \cdots \times M_n$)**具有最佳結合次序**，**求得最少的乘法運算次數，及顯示這些矩陣相乘之結合次序**，以節省計算成本。如下圖所示。

參考範例：

假設共有 3 個矩陣 M_1, M_2, M_3 要執行矩陣相乘運算，其中 M_1 為(10×100)矩陣、M_2 為(100×5)矩陣、M_3 為(5×50)矩陣，因為矩陣乘法具有結合性，所以，

$M_1\ (M_2\ M_3) = (M_1\ M_2)\ M_3$，故我們可以用兩種方法進行運算，分別求算其乘法運算次數。

1. 以 $M_1\ (M_2\ M_3)$ 進行矩陣相乘時，其乘法運算次數計算如下：

$M_{2(100 \times 5)}\ M_{3(5 \times 50)} = P_{(100 \times 50)}$ ，**所需要之乘法運算次數**：$100 \times 5 \times 50 = 25000$

$M_1\ (M_2\ M_3) = M_{1(10 \times 100)}\ P_{(100 \times 50)} = Q_{(10 \times 50)}$ ，**所需要之乘法運算次數**：$10 \times 100 \times 50 = 50000$

乘法運算次數總計為： $25000+50000 = 75000$

2. 以 $(M_1\ M_2)\ M_3$ 進行矩陣相乘時，其乘法運算次數計算如下：

$M_{1(10 \times 100)}\ M_{2(100 \times 5)} = P_{(10 \times 5)}$ ，**所需要之乘法運算次數**：$10 \times 100 \times 5 = 5000$

$(M_1\ M_2)\ M_3 = P_{(10 \times 5)}\ M_{3(5 \times 50)} = Q_{(10 \times 50)}$ ，**所需要之乘法運算次數**：$10 \times 5 \times 50 = 2500$

乘法運算次數總計為： $5000+2500 = 7500$

故**矩陣相乘次序應為** $(M_1\ M_2)\ M_3$，**且最少乘法運算次數為 7500 次**，以獲得最少之計算成本。**運算次數結果顯示在畫面上**，如下面的圖所示：