

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(例如李小明_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。
3. 人機介面的文數字一律使使用粗體及大小為 12。

試題 1:設計一連分數計算器(16 分)

說明:在數學中，連分數(continued fraction)是特殊繁分數。如果 $a_0, a_1, a_2, \dots, a_n, \dots$ 都是整數，則將分別稱為無限連分數和有限連分數。可簡記為無限項 $a_0, a_1, a_2, \dots, a_n, \dots$ 和有限項 $a_0, a_1, a_2, \dots, a_n$ 。一般一個有限連分數表示一個有理數，一個無限連分數表示一個無理數。簡單的連分數或繁分數即如圖一所示或表示為 $a_0 + 1 / (a_1 + 1 / (a_2 + \dots))$ 的數字，其中 a_0 是某個整數，而所有其他的數 a_i 都是正整數， x 是有理數分數，凡是每一個有限簡單連分數都可以化為一個有理數分數，反之亦然。用符號表示 $[a_0, a_1, a_2, \dots, a_n]$ ， $a_0, a_1, a_2, \dots, a_n$ 叫做連分數的部分商。

例子 1:連分數 $[1, 2, 3, 4]$ 經由計算可經由底部向上乘逐步計算化簡為分數，過程如圖二所示：

$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{\ddots}}}}$ <p style="text-align: center;">圖一</p>	<p style="text-align: center;">圖二</p> $1 + \frac{1}{2 + \frac{1}{3 + \frac{1}{4}}} = 1 + \frac{1}{2 + \frac{4}{13}} = 1 + \frac{13}{30} = \frac{43}{30}$
--	--

例子 2. 連分數 $[4, 2, 5, 5]$ 也可表示成運算式為 $4 + 1 / (2 + 1 / (5 + 1 / 5))$ ，其計算成為的分數過程由右向左計算如下：

$$4 + 1 / (2 + 1 / (5 + 1 / 5)) = 4 + 1 / (2 + 5 / 26) = 4 + 26 / 57 = 254 / 57$$

請設計一程式能將有限項 $[a_0, a_1, a_2, \dots, a_n]$ 之連分數計算出分數。

輸入說明：

讓使用者輸入有限項正整數 $[a_0, a_1, a_2, \dots, a_n]$ ， $1 \leq n \leq 8$ ，也就是輸入項最多 9 項、最少 2 項，

每一項的值介於 $1 \leq a_i \leq 5$ 。

輸出說明：

輸出分數。

程式執行範例如下：

注意:1. 程式需可重複執行，字型大小至少 12。2. 程式的功能，要自己寫，不可以使用現成套件或程式庫。

試題 2：0-1 背包問題 (0-1 Knapsack Problem) (17 分)

問題說明：一個小偷在商店發現了數種不同的物品，每種物品僅各有一件，並分別有其重量與價值，小偷帶了一個背包，試圖藉由選取物品以獲取最大的價值，但背包有最大的承載量之限制。由於每件物品都不可分割，因此若背包添加一個物品後超過了最大承載量，則該物品必須被捨棄(亦即，每件物品僅能不取或取，此謂之 0-1)。試設計一個程式，計算其可獲取的最大價值之金額。

w : 背包的承載量，單位為磅(pound)，最大的承載量為 W 磅。

n : 物品的總數量。

w_i : 代表編號第 i 件物品的重量，單位為磅。

v_i : 代表編號第 i 件物品的價值，單位為元(dollar)。

程式設計的說明與要求：

1. 獲取的最大價值的遞迴式(recurrence)如下：

(1) 若 $i = 0$ 或 $w = 0$ 時，總價值 $c[i, w] = c[0, x] = c[x, 0] = 0$ 。

(2) 若 $w_i > w$ 時，總價值 $c[i, w] = c[i - 1, w]$ 。

(3) 若 $i > 0$ 及 $w \geq w_i$ 時，總價值 $c[i, w] = \max(v_i + c[i - 1, w - w_i], c[i - 1, w])$ 。(例如， $\max(a, b)$ ，即從 a 與 b 兩數中取其最大者)

葉榮木 1/2/11

- 2. 本程式必須以動態規劃 (dynamic programming) 的方法解答。亦即，把一個大問題分成多個子問題，先計算各子問題的答案並儲存起來，再結合子問題的答案來計算出此大問題的答案。
- 3. 以二維矩陣的方式，逐漸地增加物品的編號 i 及背包的承載量 w ，計算並儲存 $c[i, w]$ 的數值。
- 4. 每件物品的重量與背包的承載量都可被 10 整除。(例如，10, 20, 30, ... 等)

輸入說明：

- 1. 背包的最大承載量： $W = 50$ 。
- 2. 物品的總數量： $n = 3$ 。
- 3. 各物品的重量與價值：
 $w_1 = 20, v_1 = 100; w_2 = 10, v_2 = 60; w_3 = 30, v_3 = 120$ 。

輸出說明：

- 1. 以二維矩陣呈現不同的 i 及 w 值之背包內物品的總價值 $c[i, w]$ ，其中 $c[n, W]$ 即為獲取的最大價值之金額。
- 2. 輸出格式範例： $c[0..n, 0..W]$ ，其中物品的編號 i 為 0, 1, 2, 3，承載量 w 為 0, 10, 20, 30, 40, 50。

0	0	0	0	0	0
0	x	x	x	x	x
0	x	x	x	x	x
0	x	x	x	x	x

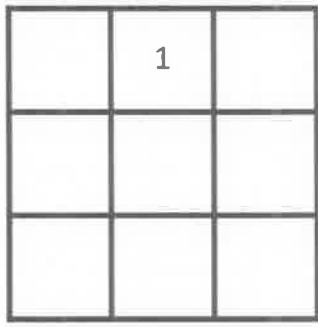
試題 3：魔方陣(16 分)

題目說明：

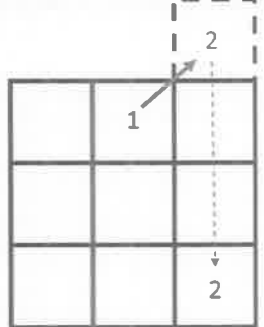
奇數魔方陣是將 1 到奇數 n 的數字排列在 $n \times n$ 的方陣，各行、各列與各對角線的和必須相同。奇數魔方陣的演算法如下：

- 1 第一個數字放在第一行第一列的中央，如圖一
- 2 數字加 1，而後向右上填數字
 - 2.1 如果填數字的格子在第一列的上方，則該數字向下沉底，如圖二。
 - 2.2 如果填數字的格子在第三行的右方，則該數字向左沉底，如圖三。
 - 2.3 如果填數字的格子已有數字，則該數字向下一格填入，如圖四。
 - 2.4 如果填數字的格子是空的，則該數字直接填入，如圖五。
 - 2.5 如果填數字的格子在第一列的上方以及第三行的右方，則該數字向下一格填入，如圖六。

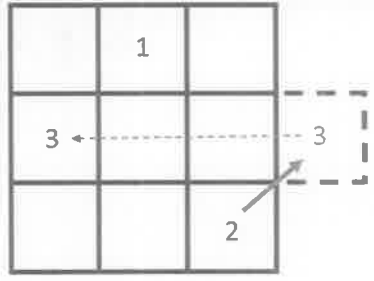
3x3 魔方陣完整的解題步驟如圖七，我們可發現各行、各列與各對角線的和均為 15。



圖一



圖二



圖三

	1	
3		
4		2

圖四

	1	6
3		5
4		2

圖五

	1	6	7
3	5	7	
4		2	

圖六

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

圖七

8	1	6
3	5	7
4	9	2

順時針旋轉90度

4	3	8
9	5	1
2	7	6

圖八

3X3 魔方陣經過順(逆)時針旋轉或左右(上下)翻轉後，其結果仍為魔方陣，圖八為順時針旋轉的範例，圖九為左右翻轉的範例。

8	1	6
3	5	7
4	9	2

左右翻轉

6	1	8
7	5	3
2	9	4

圖九

功能要求：

1. 請利用題目介紹的演算法建構一個 3X3 的魔方陣，沒有用演算法，直接將 1~9 數字填入矩陣者，不給予通過。
2. 請撰寫矩陣旋轉及翻轉的副函式，結合題卡 如圖九 的成果，建構出 3X3 魔方陣所有可能的情形，如圖十所示。

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

圖十

試題 4：英文字串壓縮程式(17 分)

題目說明：本英文字串壓縮程式將依哈夫曼編碼檔 (huffman.json)對每一個英文字母進行編碼，如右圖所示，字母 s 編碼成 0000、字母 e 編碼成 001、字母 r 編碼成 1111。此 huffman.json 的編碼是依 Jane Austen 所寫的《傲慢與偏見》一書的 ASCII 文件，其原始文字的檔案大小為 704,175 字元(字母)，相當於 5,633,400 位元。統計完所有字元 (含大小寫英文字母、數字及符號) 的頻率後，得到了本題使用的哈夫曼編碼檔；頻率高的字元編碼較短，頻率低字元編碼較長；本程式只對 88 個字元進行編碼，英文字串中若頻率高的字元居多就有較好的壓縮比。

程式執行狀態：程式不用處理輸入錯誤的問題(如輸入中文字或符號)，將 txt 檔與 exe 執行檔放在同一資料夾內。

- **輸入檔名：**程式一開始要求使用者 (你) 輸入一個 "內文為英文的檔名(.txt)"，如右圖所示，輸入 s04.txt 檔；
- **顯示該輸入檔的內容：**其內容為 ship；
- **壓縮後的編碼：**依 huffman.json 檔，輸入檔內容

的編碼碼為

0000 0001
0100
011000；

- **壓縮比：**為

0.75 (壓縮後儲存的總 Byte 數/壓縮前儲存的總 Byte 數 = 3/4)；最後不足 8bit 請分別以 0、00、000、1110、11110、111101 或 1111010 補上達 8bit，仍以 1Byte 計。本例子壓縮後的編碼檔(.bin)為 00000001 01000110 00111101。

- **儲存壓縮檔：**最後將壓縮後的 18bit 存入 s04.bin 中；其主檔名須相同於輸入檔的主檔名。

另 2 個例子如下 2 個圖：

```

1  {
2  "e": "001",
3  " ": "110",
4  "s": "0000",
5  "h": "0001",
6  "i": "0100",
7  "n": "0101",
8  "o": "0111",
9  "a": "1000",
10 "t": "1010",
11 "l": "10010",
12 "d": "10011",
13 "r": "11111",
14 "p": "011000",
15 ",": "011001",
16 ":", "011010",
17 "Z": "01101101",
18 "@": "1111010111100110000",
19 "#": "1111010111100110001",
20 "$": "1111010111100110010",
21 "%": "11110101111001100110",
22 "Q": "11110101111001100111",
23 }

```

E:\05軟體設計>113英文字串壓縮程式.exe

請輸入內文為英文的檔名(.txt): s04.txt
s04.txt 的檔案內容:
ship

壓縮後的編碼:
00000001 01000110 00
壓縮比: 0.75
壓縮後檔案名稱: s04.bin
存入壓縮檔的編碼:
00000001 01000110 00111101

E:\05軟體設計>

```
E:\05軟體設計>113英文字串壓縮程式.exe
```

```
請輸入內文為英文的檔名(.txt): s08.txt  
s08.txt 的檔案內容:  
e ee eee
```

```
壓縮後的編碼:  
00111000 10011100 01001001  
壓縮比: 0.38  
壓縮後檔案名稱: s08.bin  
存入壓縮檔的編碼:  
00111000 10011100 01001001
```

```
E:\05軟體設計>_
```

```
E:\05軟體設計>113英文字串壓縮程式.exe
```

```
請輸入內文為英文的檔名(.txt): s20.txt  
s20.txt 的檔案內容:  
she is in a hot sea.
```

```
壓縮後的編碼:  
00000001 00111001 00000011 00100010 11101000  
11000010 11110101 10000000 11000111 0100  
壓縮比: 0.50  
壓縮後檔案名稱: s20.bin  
存入壓縮檔的編碼:  
00000001 00111001 00000011 00100010 11101000  
11000010 11110101 10000000 11000111 01001110
```

```
E:\05軟體設計>_
```

繳交檢查時，除源碼檔外，請同時提供 .exe 執行檔，測試資料不限所附 3 個 .txt 檔案。

題目 5：模擬今彩 539 (17 分)

說明:根據台灣彩卷網站的說明，今彩 539 是一種樂透型遊戲，買家必須從 1~39 的號碼中任選 5 個號碼進行投注。開獎時，開獎單位將隨機開出五個號碼，這一組號碼就是該期今彩 539 的中獎號碼，也稱為「獎號」。請您依照以下演算法，寫一模擬今彩 539 隨機開出五個號碼以及相關資訊的程式。

1. 模擬今彩 539 隨機開出五個不重覆的號碼
2. 將上述五個不重覆的號碼寫入由使用者指名的文字檔案中
3. 可以由使用者指名的文字檔案中讀取五個不重覆的今彩 539 號碼
4. 計算今彩 539 五個不重覆號碼的算術平均數
5. 計算今彩 539 五個不重覆號碼的調和平均數
6. 計算今彩 539 五個不重覆號碼的幾何平均數
7. 利用選擇排序法對今彩 539 五個不重覆號碼進行由小到大的排序，進而找出今彩 539 五個不重覆號碼的中位數

算術平均數是指將所有資料的總和除以總次數，如以下式子所示：

$$\bar{x} = \frac{x_1 + x_2 + x_3 \cdots x_{n-1} + x_n}{n}$$

其中 \bar{x} 為算術平均數， x_1 為第 1 個數， $\dots x_n$ 為第 n 個數。

調和平均數(H)如以下式子所示：

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \cdots + \frac{1}{x_{n-1}} + \frac{1}{x_n}}$$

其中 x_1 為第 1 個數， $\dots x_n$ 為第 n 個數。

幾何平均數(G)如以下式子所示：

$$G = \sqrt[n]{x_1 \cdot x_2 \cdot x_3 \cdot \cdots \cdot x_{n-1} \cdot x_n}$$

其中 x_1 為第 1 個數， $\dots x_n$ 為第 n 個數。

選擇排序法是在每一輪迴圈中，都從未排序的陣列中找出最小值，跟最左側的數值交換位置，直到整個陣列排序完畢。

中位數是指，將一組數值資料由小到大排列後，最中間的數值即為中位數。

華榮木

輸入說明：

~~可以讓使用者輸入無人機飛行操控命令檔案(假設 input.txt)，此命令檔案內容。~~

1. 讓使用者輸入要**寫入**今彩 539 五個不重覆號碼的文字檔
2. 讓使用者輸入要**讀取**今彩 539 五個不重覆號碼的文字檔

輸出和功能要求：

1. 能正確隨機開出五個**不重覆**的號今彩 539 碼
2. 能正確將今彩 539 五個不重覆號碼**寫入**使用者輸入的文字檔中
3. 能正確讀入使用者輸入的文字檔中之今彩 539 五個不重覆號碼
4. 能正確算出今彩 539 五個不重覆號碼的算術平均數
5. 能正確算出今彩 539 五個不重覆號碼的調和平均數
6. 能正確算出今彩 539 五個不重覆號碼的幾何平均數
7. 能展示選擇排序法之排序過程
8. 能正確算出今彩 539 五個不重覆號碼的中位數

程式執行範例：

```
***模擬今彩539***
今彩539之5個1~39號碼: 4 28 11 38 2
請輸入存今彩539之5個號碼的檔名: a539.txt

請輸入要讀今彩539之5個號碼的檔名: a539.txt
今彩539之5個號碼的算數平均數: 16.600000
今彩539之5個號碼的調和平均數: 5.537471
今彩539之5個號碼的幾何平均數: 9.869267

排序前的資料: 4 28 11 38 2

第 1 次選擇: 2 28 11 38 4*
               --
第 2 次選擇: 2 4 11 38 28*
               -- --
第 3 次選擇: 2 4 11* 38 28
               -- --
第 4 次選擇: 2 4 11 28 38*
               -- --

排序後的資料: 2 4 11 28 38

今彩539之5個號碼的中位數: 11
```

此範例隨機產生今彩 539 五個不重覆號碼: 4 28 11 38 2，寫入 a539.txt 中，從 a539.txt 中讀取今彩 539 五個不重覆號碼，其算術平均數為 16.6，調和平均數為 5.537471，幾何平均數為 9.869267，在選擇排序過程中，第 1 次從 5 個數(4 28 11 38 2)中選擇最小值 2，跟最左邊的 4 交換，*表示交換，2 的下方__表示數字 2 已經**排序好了**。第 2 次從未排序的 4 個數(28 11 38 4)中選擇最小值 4，跟未排序最左邊的 28 交換，依此類推，最後排序結果為 2 4 11 28 38。其中位數為 11。

程式的功能，要自己寫，不可以使用現成套件或程式庫，若妳(你)的程式都完成上述功能要求，才可以上傳檢查。

試題 6：影像處理之邊緣偵測實作(17 分)

說明：在影像處理中，邊緣偵測是基礎且重要的關鍵技術。其有助於提取影像中的形狀或物體輪廓特徵，因而被廣泛用於自動駕駛、工業瑕疵檢測、醫學影像分析等領域。Sobel 是常見的邊緣偵測技術

之一，其演算法如下：

1. 輸入一張彩色影像(RGB Image)，其大小為高度(H)×寬度(W)×3。每個像素由紅色 (R)、綠色 (G)、藍色 (B) 三種顏色組成，每種顏色分別用 8 位元表示。
2. 將彩色影像轉換為灰階影像 (Gray Image)，這是許多影像處理技術的基礎步驟。使用如下的轉換公式：

$$Gray(x,y) = 0.3 \times R(x,y) + 0.59 \times G(x,y) + 0.11 \times B(x,y) \quad (1)$$

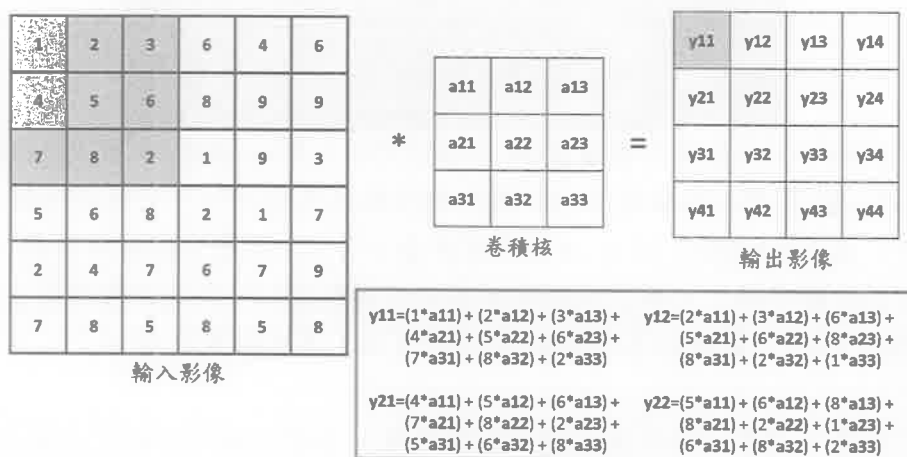
3. 利用 Sobel 邊緣偵測方法對灰階影像進行處理。使用水平及垂直 Sobel 運算子進行卷積操作，計算水平和垂直邊緣的梯度。其中，水平(x)方向梯度 G_x 如方程式(2)所示。

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * Gray \quad (2)$$

垂直方向(y)梯度 G_y 則如方程式(3)所示：

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * Gray \quad (3)$$

其中，‘*’表示卷積(Convolution)操作，可以理解為一種「滑動窗口」的操作過程。首先將一個小型矩陣(即卷積核)覆蓋在影像的某一個區域上，通常從影像的左上角開始。如圖一所示，卷積核與該區域的每個像素值相乘，再將這些乘積加總，此加總結果將成為新影像在該位置的像素值。接著我們將卷積核向右移動一個像素位置，重複相同的計算過程。當移動到一行的末尾後，卷積核會移動到下一行的起始位置，繼續執行相同的操作。這樣逐像素地掃描整個影像，最終可獲得經卷積處理的新影像。



圖一、卷積操作

4. 在獲得 G_x 與 G_y 後，藉由方程式(4)可進一步計算得出最終邊緣偵測結果：

$$G = \sqrt{G_x^2 + G_y^2} \quad (4)$$

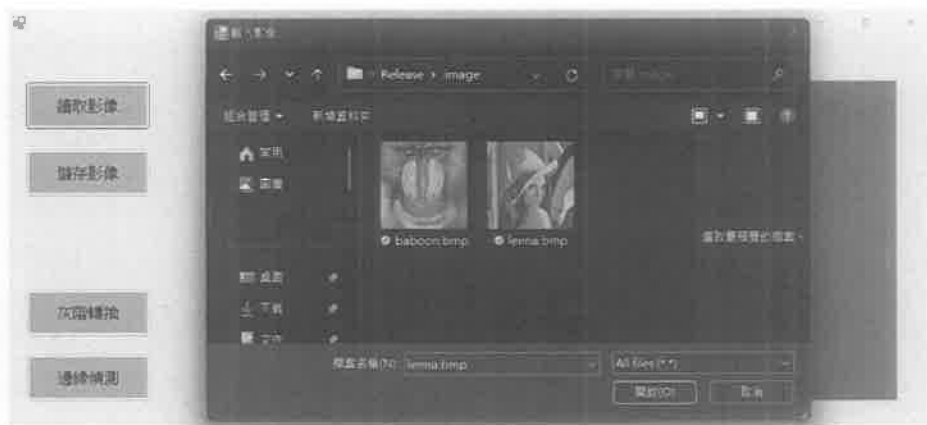
程式功能:

請寫一個圖形使用者介面(GUI)，並利用上述演算法，完成以下功能要求：

1. 使用者能選取要輸入的影像圖檔，可選擇包括(jpg, png, bmp, tif 等)。
2. 使用者能點選功能按鈕，將輸入的彩色影像轉換為灰階影像。
3. 使用者能點選功能按鈕，將輸入的彩色影像轉換為邊緣偵測後的影像。
4. 使用者能依照自己的需求自由命名檔案及選擇格式(jpg, png, bmp, tif)，並儲存當前輸出結果，包括灰階影像與邊緣偵測後的影像。

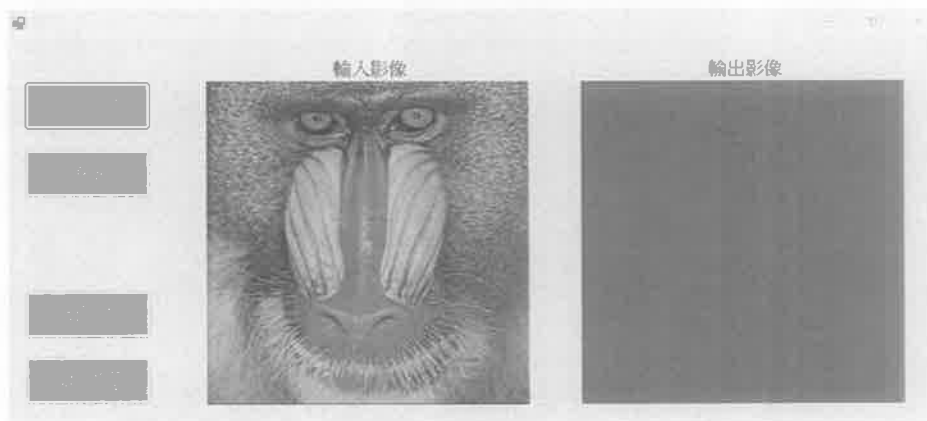
程式執行範例：

1. 當程式執行時，會出現圖形使用者介面，如圖二所示。使用者可點選讀取影像按鈕來選擇輸入影像。



圖二、讀取影像示意圖

2. 當使用者開啟彩色影像檔後，程式會自動顯示，如圖三所示。



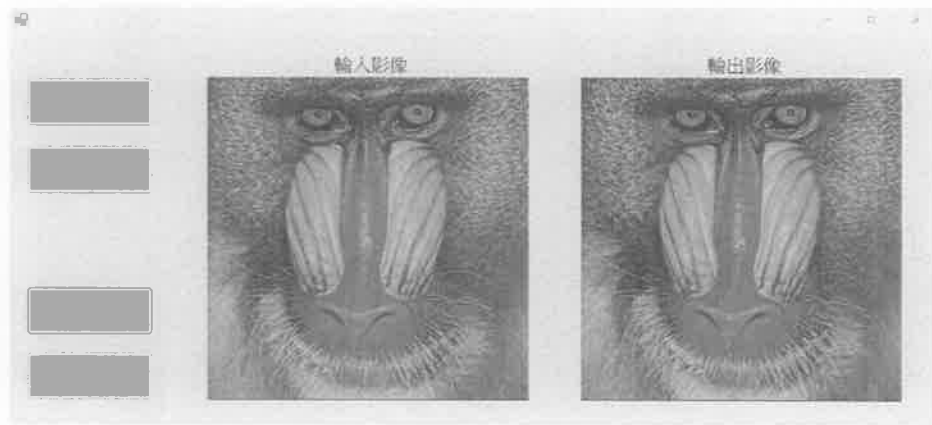
(a)



(b)

圖三、讀取影像結果示意圖: (a) 範例一; (b) 範例二

3. 當使用者點選灰階轉換功能按鈕時，彩色影像將轉換成灰階影像，並顯示於右側，如圖四所示。



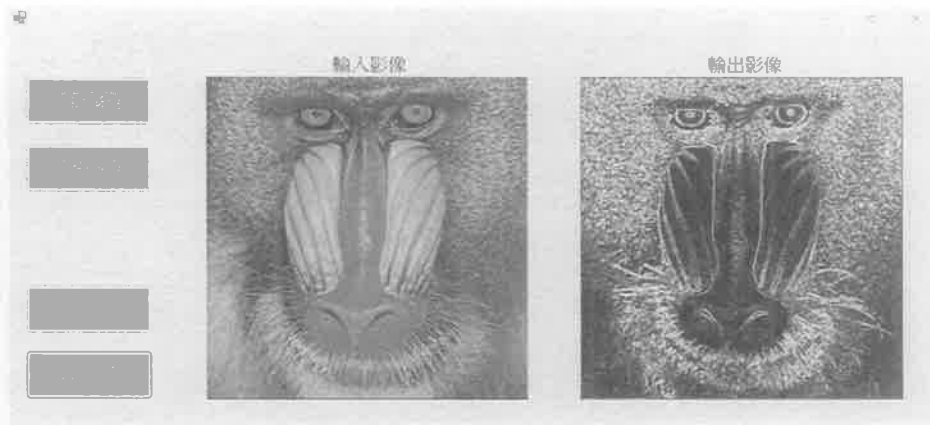
(a)



(b)

圖四、灰階轉換結果示意圖: (a) 範例一; (b) 範例二

4. 當使用者點選邊緣偵測按鈕功能時，彩色影像將轉換成邊緣偵測後的影像，並顯示於右側，如圖五所示。



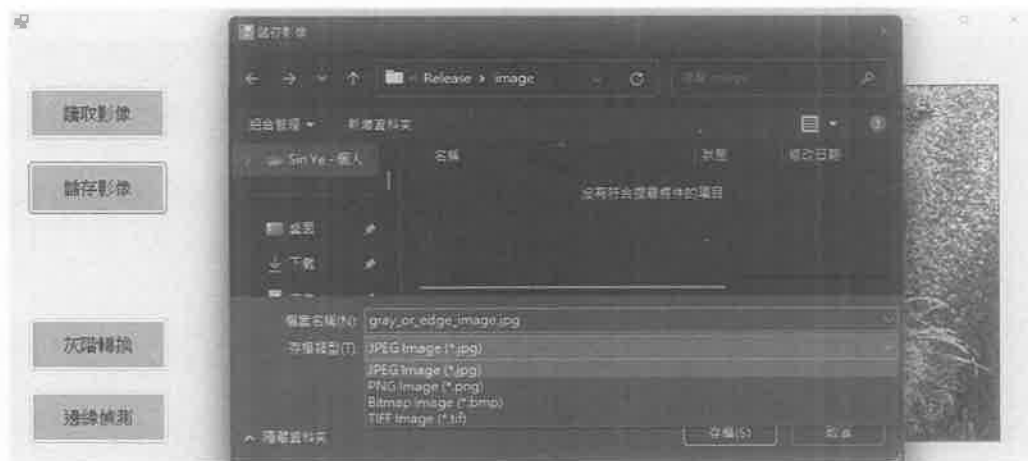
(a)



(b)

圖五、邊緣偵測結果示意圖: (a) 範例一; (b) 範例二

5. 使用者在操作動作 3 及動作 4 後，可依照需求，點選儲存按鈕，儲存當前轉換後的影像(灰階或邊緣影像)，其可自由地命名及儲存不同影像格式(jpg, png, tif, bmp)，如圖六所示。



(a)



(b)

圖六、儲存影像示意圖: (a) 範例一; (b) 範例二