

台北市 101 學年度高級中等學校學生（高工組）電腦軟體設計競賽 決賽試題

工作桌編號 _____ 選手姓名 _____ 代表學校 _____ 總分 _____

試卷說明:1. 請將寫好之程式原始檔依題號命名存檔，第一題取：選手姓名_Q1，第二題取：選手姓名_Q2，依序命名存檔，並存於 C 碟之選手姓名_Contest 目錄。2. 競賽時間 4 小時。

試題一：解工程合作問題

說明：試設計一個程式，可用來解下列工程合作問題。

有一件工程，由甲獨做，需要 X 天完成，由乙獨做，需要 Y 天完成，若已知下列條件，求 X 和 Y ？

條件 1: 甲、乙合作 N_1 天可完成。

條件 2: 由甲、乙先合作 N_2 天後，再由甲獨作 N_3 天，剩下的再由乙獨作 N_4 天。

條件 3: N_3 不等於 N_4 ，且 X, Y, N_1, N_2, N_3 ，及 N_4 的值都是整數。

評分項目：

1. 可以正確的解出 X 與 Y 的值(如圖一所示)。(20 分)
2. 當無解時，可以在答案區顯示“無解”(如圖二所示)。(5 分)

解工程合作問題

有一件工程，由甲獨做需要 X 天完成，由乙獨做需要 Y 天完成，若已知下列條件，求 X 和 Y ？

1. 甲、乙合作 $N_1 = 24$ 天可完成。

2. 由甲、乙先合作 $N_2 = 6$ 天，再由甲獨作 $N_3 = 15$ 天，剩下的再由乙獨作 $N_4 = 20$ 天。

3. N_3 不等於 N_4 ，且 X, Y, N_1, N_2, N_3 ，及 N_4 的值都是整數。

解:

圖一

解工程合作問題

有一件工程，由甲獨做需要 X 天完成，由乙獨做需要 Y 天完成，若已知下列條件，求 X 和 Y ？

1. 甲、乙合作 $N_1 = 24$ 天可完成。

2. 由甲、乙先合作 $N_2 = 5$ 天，再由甲獨作 $N_3 = 10$ 天，剩下的再由乙獨作 $N_4 = 20$ 天。

3. N_3 不等於 N_4 ，且 X, Y, N_1, N_2, N_3 ，及 N_4 的值都是整數。

解:

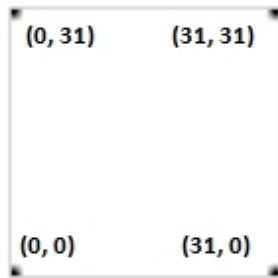
圖二

題目二：框選圖中的阿拉伯數字

背景：在進行攝影或影像監控時，均需對圖片或影片中的物件進行框選，再做後續處理。請根據所提供的數字 0~9 圖形檔案，進行實作。

實作：為簡化程式的撰寫，僅以 32x32 的 BMP 圖檔為實作對象，BMP 圖上任一點在 BMP 圖檔上是用 3Bytes(藍、綠、紅)存放，若是白點則存放著 255、255、255，若是黑點則存放著 0、0、0。本題目所附的 BMP 圖檔是由黑色的點組成。

本題目特別規定 BMP 圖左下角座標為(0, 0)、右下角座標為(31, 0)、左上角座標為(0, 31)、右上角座標為(31, 31)，如下左圖所示。



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0000h:	42	4D	36	0C	00	00	00	00	00	00	36	00	00	00	28	00
0010h:	00	00	20	00	00	00	20	00	00	00	01	00	18	00	00	00
0020h:	00	00	00	0C	00	00	13	0B	00	00	13	0B	00	00	00	00
0030h:	00	00	00	00	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF
0040h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0090h:	FF	FF	FF	00	00	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0bd0h:	FF	FF	FF	FF	FF	FF	00	00	00	FF	FF	FF	FF	FF	FF	FF
0c30h:	FF	FF	FF	00	00	00										

BMP 圖檔的第 0、1 位元組一定是存放 42h、4Dh (h 表十六進位)，如上右圖所示。

第 12h~15h 位元組存放圖面寬的點數(32 點)，第 16h~19h 位元組存放圖面高的點數(32 點)。

第 0036h~38h 位元組存放點座標(0, 0)的藍、綠、紅之值為 0、0、0 表示一個黑點。

第 0039h~3bh 位元組存放點座標(1, 0)的藍、綠、紅之值為 255、255、255 表示一個白點。

第 0093h~95h 位元組存放點座標(31, 0)的藍、綠、紅之值為 0、0、0 表示一個黑點。

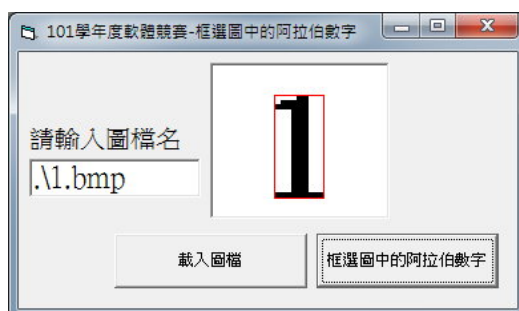
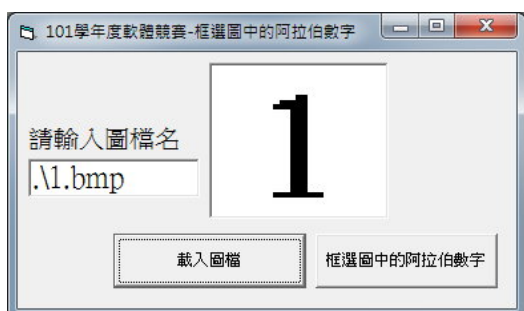
第 0bd6h~0bd8h 位元組存放點座標(0, 31)的藍、綠、紅之值為 0、0、0 表示一個黑點。

第 0c33h~0c35h 位元組存放點座標(31, 31)的藍、綠、紅之值為 0、0、0 表示一個黑點。

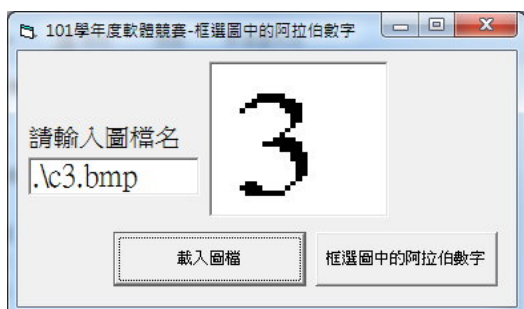
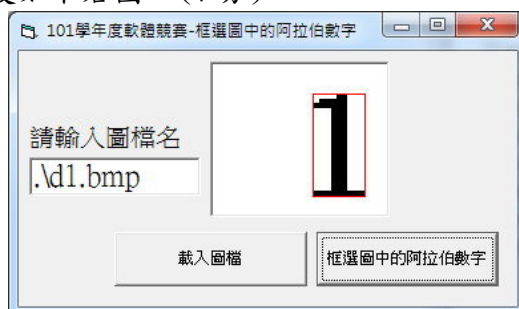
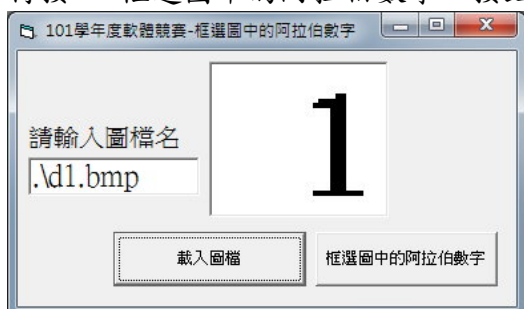
請寫一支程式能框選 32x32 的 BMP 圖中的阿拉伯數字。參考介面如下圖所示。(3 分)



例子一：分別載入 0.bmp ~ 9.bmp 檔案，按了 “載入圖檔” 按鈕後如下左邊圖，再按 “框選圖中的阿拉伯數字” 按鈕後如下右邊圖。(15 分)



例子二：載入位置不同的圖檔，如下左邊圖，再按 “框選圖中的阿拉伯數字” 按鈕如下右圖。(7 分)



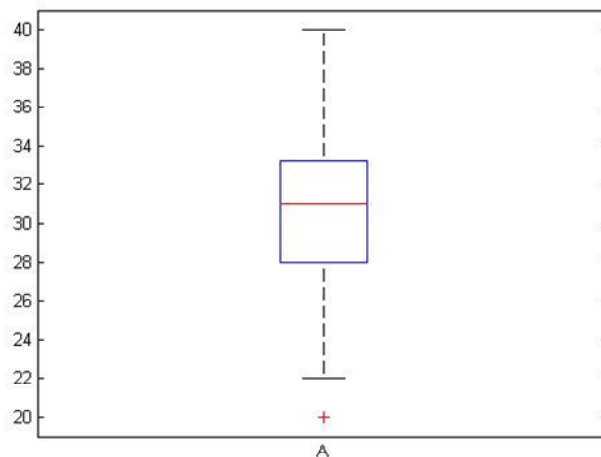
試題三：畫盒鬚圖 (Box Plot)

說明：盒鬚圖常被用於，提供有關資料的位置和分散的參考，尤其在不同的母體資料時，更可以表現其差異，假設有 n 筆資料，其繪製須有以下的統計量：

1. 資料的最小值和最大值。
2. 中位數(Median): 樣本資料經由遞增或遞減的排序後，若 n 是奇數，則中位數為中間值。若 n 是偶數，則中位數為中間兩個值的平均數。
3. 百分位數: 第 p 個樣本百分位數是某一個數值 d_p ，使得樣本中有 p 部分的觀察值小於或等於 d_p 。求第 p 個樣本百分位數的方法: (1) 將資料由小到大排序。(2) 計算第 $i = \left(\frac{p}{100}\right)n$ 個位置(3)若 i 不為整數，則取下一個 i 還要大的值，即為第 p 個樣本百分位數。若 i 為整數，則第 p 個樣本百分位數為第 i 和 $i+1$ 的平均。注意：第 50 個百分位數=中位數。
4. 四分位數：將資料區分為 4 個部份，這些區分的點即稱為四分位數，包括第一四分位數(Q1)或第 25 個百分位數，第二四分位數(Q2)或第 50 個百分位數，第三四分位數(Q3)或第 75 個百分位數。
5. 四分位距(IQR)：即為第一個四分位數和第三個四分位數的差， $IQR = Q3 - Q1$ 。
6. 最小下限： $Q1 - IQR$
7. 最大上限： $Q3 + IQR$
8. 異常值(Outliers)：資料的值，小於最小下限或大於最大上限

請你設計一程式，可以讓使用者輸入 n 筆資料，然後畫出這些資料的盒鬚圖，同時輸出這些資料的最大值，最大上限，IQR，Q3，Median，Q1，最小下限以及 Outlier 等值。執行範例如下：

假設 A 班有 13 位同學的考試成績如下: 39, 32, 20, 34, 40, 33, 31, 29, 25, 30, 31, 32, 22，其盒鬚圖如下，其中 Median 畫紅色實線(寬度 20)，Outlier 畫紅色+，IQR 畫黑色虛線，最大上限和下限畫出黑色實線(寬度 10)，Q1 和 Q3 之間的矩形(寬度 20)，用藍色實線畫。水平顯示那一類的資料，以下圖為例，即是 A 類資料。垂直顯示由 Min 到 Max 之間資料。下圖中之 Max, Min, IQR, 最大上限, Q3, Q1, 最小下限以及離群值(Outlier)，是說明用，在您的程式中，不需要顯示。



評分：

1. 程式介面 (2 分)。
2. 可以讓使用者輸入資料(2 分)。
3. 可以劃出水平座標(2 分)。
4. 可以劃出垂直座標，且會隨著輸入資料不同而變動(5 分)。
5. 輸出資料的最大值，最小值，最大上限，IQR，Q3，Median，Q1，最小下限以及 Outlier 等值(4 分)。
6. 劃出的盒鬚圖正確(10 分)。

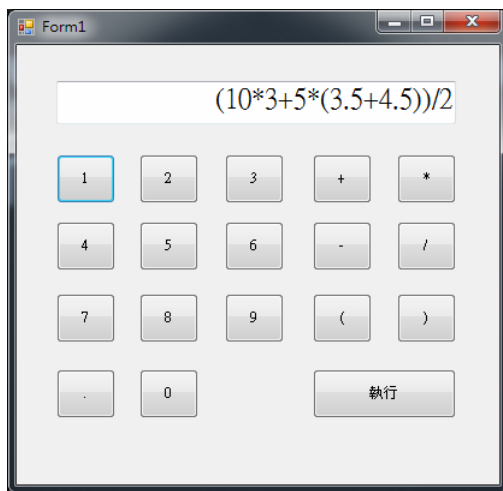
題目四：四則運算機

說明：

設計一簡易型計算機介面，使用者從該介面輸入四則(加減乘除)運算式後，程式需將原四則運算式與正確結果輸出至檔案(b.txt)中。輸入之四則運算式中需包含有括號以改變運算之優先順序，且可輸入不只一層括號，輸入之數值需可包含浮點數。如輸入之四則運算式含有其他無法辨識的符號(例如 a、c、\$、%、... 等)，或運算式不合規範(括號數錯誤、括號順序錯誤、運算子錯誤、浮點數格式錯誤、... 等)，則程式需顯示” 運算式有誤!” 字樣，且不會產生 b.txt 檔案。

參考範例一：

輸入四則運算式： $(10*3+5*(3.5+4.5))/2$



執行後，程式會自動產生一 b.txt 檔案，並於 b.txt 檔案中輸出： $(10*3+5*(3+5))/2 = 35$

註：b.txt 檔案輸出之方程式中，”=” 前後需各含有一個空格。

參考範例二：

輸入四則運算式： $(2.5*4-2+5\%(2.7+5.3))/2$

程式執行後，系統顯示” 運算式有誤!” 字樣，且不會產生 b.txt 檔案。

提示：

平常所使用的運算式，主要是將運算元放在運算子的兩旁，例如： $a+b/d$ ，這樣的式子，這稱之為中序 (Infix) 表示式。對於人類來說，這樣的式子很容易理解，但由於電腦執行指令時是有順序的，遇到中序表示式時，無法直接進行運算，而必須進一步判斷運算的先後順序，所以必須將中序表示式轉換為後序 (Postfix) 表示式，後序表示式又稱之為逆向波蘭表示式 (Reverse polish notation)。例如： $(a+b)*(c+d)$ 這個式子，後序表示式為： $ab+cd+*$ ，轉換演算法的輸出過程如下：

OP	STACK	OUTPUT
((-
a	(a
+	(+	a
b	(+	ab
)	-	ab+
*	*	ab+
(*(ab+
c	*(ab+c
+	*(+	ab+c
d	*(+	ab+cd
)	*	ab+cd+
-	-	ab+cd+*

評分：

1. 程式介面(5 分)
2. 程式執行正確性(20 分)