

第 51 屆全國技能競賽分區技能競賽

競賽試題(說明)

(註：本公開試題共 8 題，於競賽時為其中的 4 題且內容有百分之三十之調整)

第 1 題：最大公因數計算(配分 15 分)

1. 題目說明

請開啟 **CSD01.csproj** 專案的 Program.cs 檔案，用 while 迴圈，完成以下要求，要求使用者輸入兩個大於等於 1 而且小於 100 的數字，最後尋找最大公因數。請依下列題意完成作答，儲存完整程式，產出的執行檔命名為 **CSA01.exe**。

2. 設計說明：

- (1) 主程式中輸入兩個介於 1-100 的整數，分別為 input1 和 input2，判斷兩數是否皆大於等於 1 而且小於 100，否則顯示【輸入錯誤，請重新執行】。
- (2) 宣告一個整數變數 result，儲存 input1 和 input2 的最大公因數。
- (3) 建立 findCommon 函式，並於函數內部使用 while 迴圈，傳入兩個整數，傳出此兩個整數之最大公因數。
- (4) 以 try...catch 判斷，若輸入非數字，則顯示【輸入錯誤，請重新執行】，並中斷執行程式。
- (5) 最終運用格式化輸出結果，如【input1 和 input2 的最大公因數是 xx】。

3. 執行結果參考畫面：

- (1) 於函數內部使用 while 迴圈，傳入兩個整數，傳出此兩個整數之最大公因數。



```
請輸入第一個介於1-100的整數>90
請輸入第二個介於1-100的整數>33
90和33的最大公因數是3
```

- (2) 以 try...catch 判斷，若輸入非數字，則顯示【輸入錯誤，請重新執行】，並中斷執行程式。



```
請輸入第一個介於1-100的整數>99.B
輸入錯誤，請重新執行
```

第 2 題：迴圈運算(配分 15 分)**1. 題目說明**

請開啟 **CSD01.csproj** 專案的 Program.cs 檔案，由鍵盤輸入兩個整數以及一個運算子（如： $+$ 、 $-$ 、 $*$ 、 $/$ ），判斷運算子並計算對應結果後輸出。請依下列題意完成作答，儲存完整程式，產出的執行檔命名為 **CSA01.exe**。

2. 設計說明：

- (1) 使用 Console.Write 與 Console.ReadLine 迴圈，要求使用者分別輸入兩組整數。提示訊息【請輸入第一個整數>】與【請輸入第二個整數>】。以 try...catch 判斷，若輸入非數字，則顯示【輸入錯誤，請重新執行】，並中斷執行程式。
- (2) 使用 Console.Write 與 Console.ReadLine 迴圈，要求使用者輸入一個運算子，可能輸入為 $+$ 、 $-$ 、 $*$ 、 $/$ ，顯示提示訊息【請輸入運算子(+*/>】（+*/>）皆為半型。若非以上四者，則提示【運算子必須為+*/之一】，且需重新輸入， $+$ 、 $-$ 、 $*$ 、 $/$ 皆必須可以執行。
- (3) 運算子重複輸入的部分必須使用 do...while，計算數學運算部分則必須使用 if... else...。
- (4) 依輸入的兩個數字及運算子，計算對應結果後輸出。
* 例： 輸入 5、2、+ → 顯示 5+2=7。
輸入 3、8、* → 顯示 3*8=24。
輸入 58、2、/ → 顯示 58/2=29。
輸入 9、19、- → 顯示 9-19=-10。

3. 執行結果參考畫面：

- (1) 由鍵盤輸入兩個整數以及一個運算子（如： $+$ 、 $-$ 、 $*$ 、 $/$ ），判斷運算子並計算對應結果後輸出。

```
請輸入第一個數字>5
請輸入第二個數字>2
請輸入運算子(+*/>)+
5+2=7
```

```
請輸入第一個數字>9
請輸入第二個數字>19
請輸入運算子(+*/>)-
9-19=-10
```

- (2) 以 try...catch 判斷，若輸入非數字，則顯示【輸入錯誤，請重新執行】，並中斷執行程式。

```
請輸入第一個數字>29
請輸入第二個數字>bbbbbb
輸入錯誤，請重新執行
```

- (3) 若運算子非輸入+*/四者之一，則提示【運算子必須為+*/之一】，且需重新輸入。

```
請輸入第一個數字>90
請輸入第二個數字>88
請輸入運算子(+*/>)zzz
運算子必須為+*/之一
請輸入運算子(+*/>)_
```

第 3 題：賓果遊戲(配分 15 分)**1. 題目說明**

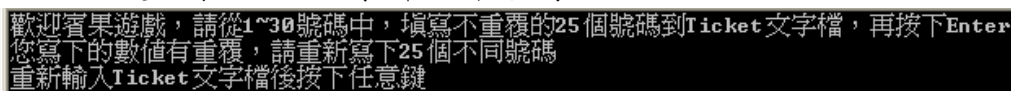
請開啟 **CSD02.csproj** 專案的 Program.cs 檔案，使用文字檔輸入 5x5 的 25 格的數字，數字區間為 1~30 數值，電腦會隨機開出 25 個數字以比對使用者輸入的數字，若在相同位置有相同的數字，則為中獎號碼，程式可進行比對使用者共連出幾條線。請依下列題意完成作答，儲存完整程式，產出的執行檔命名為 **CSA02.exe**。

2. 設計說明：

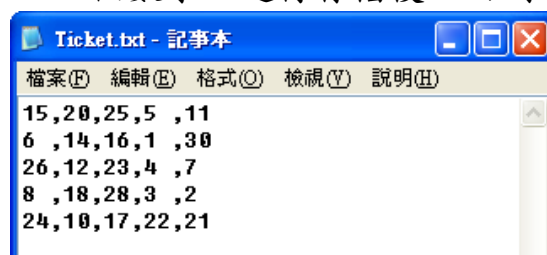
- (1) 使用專案 Ticket.txt 提供使用者寫下數字，接著讀取檔案的數字放到多維度陣列，若取出的數字有誤或重複，則顯示提示訊息【您寫下的數值有重複，請重新寫下 25 個不同號碼】及【重新輸入 Ticket 文字檔後按下任意鍵】，重新讓使用者再重新輸入一次 Ticket.txt，直到 25 個數字都正確取得。
- (2) 使用 BingoGame 類別 bet 函式，判斷玩家的數字是否有重複、空值或是大於 30 以上的數字。
- (3) 使用 BingoGame 類別 Bingo 方法，開獎判斷使用者總共連線數。
- (4) 使用 BingoGame 類別 GetLottery 函式顯示開獎號碼，同時比對使用者所寫的號碼，中獎號碼使用黃色顯示，未中獎號碼則使用白色顯示。

3. 執行結果參考畫面：

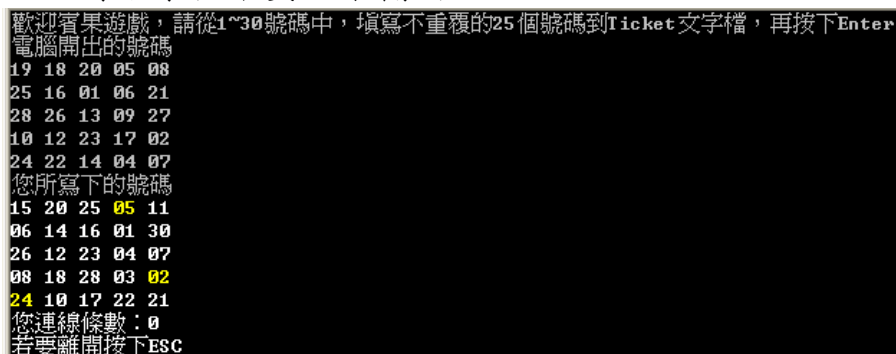
- (1) 若取出的數字有誤或重複，則顯示提示訊息，重新讓使用者再重新輸入一次 Ticket.txt，直到 25 個數字都正確取得。



- (2) 開啟 Ticket.txt 輸入 25 個數字，進行存檔後，回到主畫面按下 Enter 鍵。



- (3) 系統開始比對數字把中獎結果輸出。



第 4 題：轉檔程式(配分 15 分)**1. 題目說明**


請開啟 **CSD02.csproj** 專案的 Program.cs 檔案，使用 System.Convert 類別把開啟圖檔轉換為使用 Base-64 位數編碼的相等字串另存文字檔，也可以把轉換的 Base-64 文字檔開啟轉換為相等的 8 位元不帶正負號的整數陣列另存成新圖檔。請依下列題意完成作答，儲存完整程式，產出的執行檔命名為 **CSA02.exe**。

2. 設計說明：

- (1) 程式執行時，顯示提示文字等待使用者按鍵輸入，若輸入 1(以鍵盤或數字鍵)，則進行圖檔轉成文字檔作業；若輸入 2(以鍵盤或數字鍵)，則進行文字檔轉圖檔作業；按下 Esc 離開此轉檔程式。
- (2) 按下 1，進行圖檔轉文字檔作業，先判斷來源檔案是否存在，若檔案不存在，顯示提示訊息【此圖檔不存在！】；若檔案存在，要求輸入欲轉檔的圖檔來源，使用 EncodeFile 方法進行轉檔，檔名為使用者設定的轉出檔案名稱。
- (3) 按下 2，進行文字檔轉圖檔作業，先判斷來源檔案是否存在，若檔案不存在，顯示提示訊息【此轉檔不存在！】；若檔案存在，要求輸入欲轉檔的文字檔來源，使用 DecodeFile 方法進行轉檔，檔名為使用者設定的轉出圖檔名稱。
- (4) 按下鍵盤 Esc 鍵，離開程式。


3. 執行結果參考畫面：

- (1) 程式執行後畫面會等待使用者按鍵輸入。

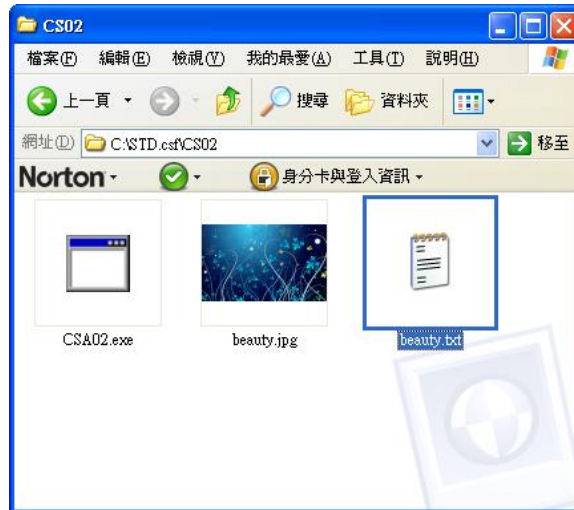


```
轉檔程式：
按 1 將圖檔轉成文字檔
按 2 將文字檔轉成圖檔
按 Esc 離開此程式
```

- (2) 按下 1，進行圖檔轉文字檔作業，先判斷來源檔案是否存在，若檔案不存在，顯示提示訊息【此圖檔不存在！】；若檔案存在，要求輸入欲轉檔的圖檔來源，並依使用者設定的轉出檔案名稱進行命名及轉檔。



```
請輸入圖檔來源：
aaa.jpg
此圖檔不存在！
請輸入圖檔來源：
beauty.jpg
請輸入轉出檔案名稱：
beauty.txt
轉檔成功！按下任意鍵跳出
```



- (3) 按下 2，進行文字檔轉圖檔作業，先判斷來源檔案是否存在，若檔案不存在，顯示提示訊息【此轉檔不存在！】；若檔案存在，要求輸入欲轉檔的文字檔來源，並依使用者設定的轉出圖檔名稱進行命名及轉檔。

```
請輸入轉檔來源：
ccc.txt
此轉檔不存在！
請輸入轉檔來源：
beauty.txt
請輸入轉出圖檔名稱：
newphoto.jpg
轉檔成功！按下任意鍵跳出
```



第 5 題：撲克牌洗牌程式(配分 30 分)**1. 題目說明**

請開啟 **CSD04.csproj** 專案的 Program.cs 檔案，實作一個撲克牌洗牌程式。請依下列題意完成作答，儲存完整程式，產出的執行檔命名為 **CSA04.exe**。

2. 設計說明：

- (1) 撲克牌每副 52 張，分別為梅花 (Clubs)、方塊 (Diamonds)、紅心 (Hearts) 與黑桃 (Spades)，每張花色有 13 張牌 (1-10、J、Q、K)，程式必須進行洗牌，使得每次的順序及花色皆不相同。

* 提示：可使用 Random 類別。

- (2) 自洗完後的牌中隨機抽出五張牌並輸出，輸出的內容必須有花色以及牌上的號碼，不可出現重複的撲克牌。

* 提示：假設從 52 個號碼中隨機抽取號碼，依此號碼對應以下花色（如隨機抽出 25，則代表為方塊 Q；抽出 35，則代表為紅心 9）。

號碼	1~13	14~26	27~39	40~52
花色	梅花 1~13	方塊 1~13	紅心 1~13	黑桃 1~13

3. 執行結果參考畫面：

```
No.1, Clubs 4
No.2, Hearts 3
No.3, Clubs 6
No.4, Diamonds K
No.5, Clubs 5
```

第 6 題：快速排序法(配分 30 分)**1. 題目說明**

請開啟 **CSD04.csproj** 專案的 Program.cs 檔案，編寫快速排序法 (Quick Sort) 以排序數列請依下列題意完成作答，儲存完整程式，產出的執行檔命名為 **CSA04.exe**。

2. 設計說明：**(1) 快速排序法的步驟如下：**

- a. 從數列中挑出一個元素，稱為基準值 (pivot)。
- b. 重新排序數列，所有元素比基準值小的擺放在基準值前面，所有元素比基準值大的擺在基準值的後面(相同的數可以到任一邊)。在這個分割結束之後，該基準就處於數列的中間位置。這個稱為分割 (partition) 操作。
- c. 遞迴地 (recursive) 把小於基準值元素的子數列和大於基準值元素的子數列排序。

(2) 程式內已給予 15 個數值，類別需傳入 `IEnumerable<int>` 型別的數列集合，以及 `Sort()` 方法傳回 `IEnumerable<int>` 型別的已排序數列。

3. 執行結果參考畫面：

```
032
043
054
055
074
084
094
096
132
391
456
531
563
652
726
```

第 7 題：訂房系統(配分 40 分)**1. 題目說明**

請開啟 **COD02.sln** 方案的 Form1.cs 檔案，完成「訂房系統」的功能，使用者可以選擇入住日期、住房人數與付款方式並進行預訂或取消。請依下列題意完成作答，儲存完整程式，並製成執行檔 **COA02.exe**。

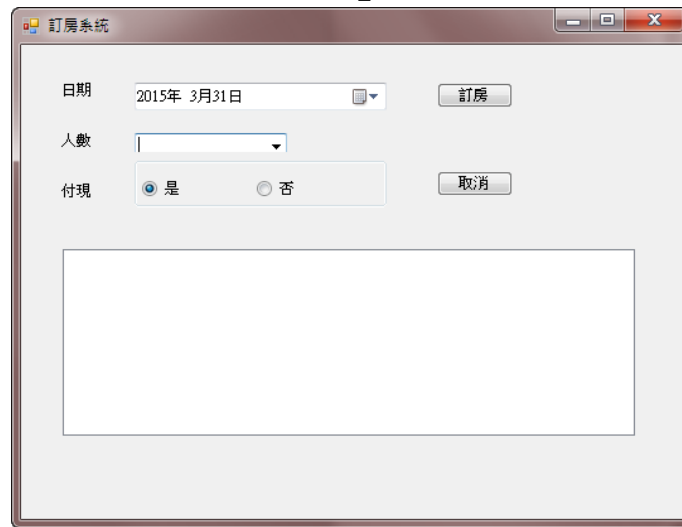
<參考圖>

2. 設計說明：

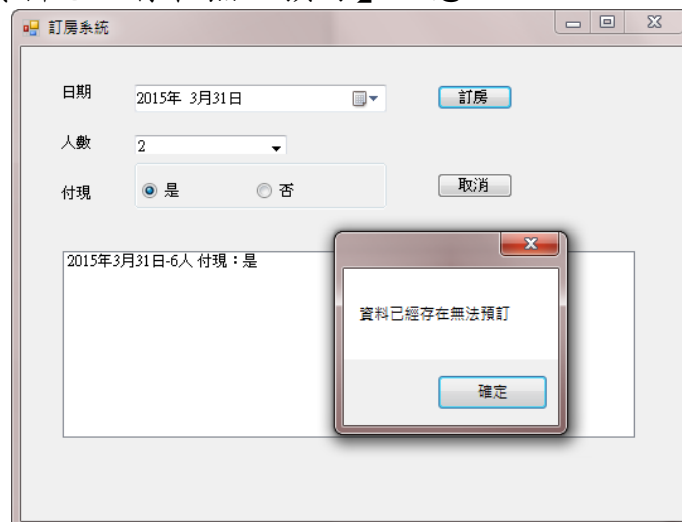
- (1) 表單載入時，付現的預設值為「是」。
- (2) 預約訊息儲存於 Dictionary<String, String> record 之中。key 為日期，value 為記錄，所有預約訊息顯示於 ListBox 中如格式：key "-" value，value 記錄格式為：人數（一個半形空白）付現：是（或否）。
顯示如：2014 年 11 月 12 日-2 人 付現：是。
- (3) 點選「訂房」按鈕，新增一筆記錄顯示下方 ListBox 列表資料中，並顯示【訂房完成】訊息。一天只能預約一筆資料，若選擇的日期已經有記錄則無法預約，顯示【資料已經存在無法預訂】訊息；若未選擇人數即訂房，顯示【請選擇人數】訊息。
- (4) 點選「取消」按鈕，若選擇日期沒有記錄，則無法取消，顯示【查無記錄】訊息；否則刪除該筆資料，並重新顯示 ListBox 的列表資料。
- (5) 資料更新時，必須重新顯示 ListBox 的列表資料。

3. 執行結果參考畫面：

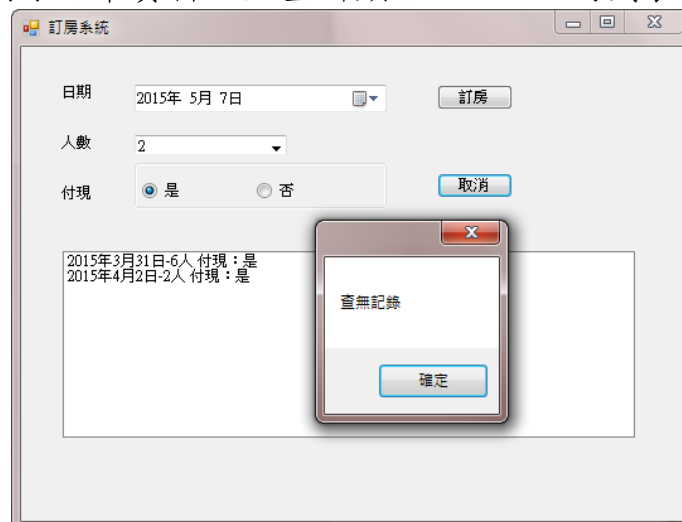
(1) 表單載入時，付現的預設值為「是」。



(2) 點選「訂房」按鈕，新增一筆記錄顯示下方 ListBox 列表資料中，並顯示【訂房完成】訊息。一天只能預約一筆資料，若選擇的日期已經有記錄則無法預約，顯示【資料已經存在無法預訂】訊息。



(3) 點選「取消」按鈕，若選擇日期沒有記錄，則無法取消，顯示【查無記錄】訊息；否則刪除該筆資料，並重新顯示 ListBox 的列表資料。



第 8 題：產品銷售表(配分 40 分)**1. 題目說明**

請開啟 **COD03.sln** 方案的 Form1.cs 檔案，完成「產品銷售表」的功能，使用專案資料庫透過 Entity Framework 來存取資料庫內容，執行預存程序，並將每個執行資料庫的命令都存到文字檔。請依下列題意完成作答，儲存完整程式，並製成執行檔 **COA03.exe**。

產品	進貨日期	進貨數量	銷售數量
奮起湖便當	2014/10/3	10	8
新國民便當-排骨	2014/10/2	20	8
新國民便當-油雞	2014/10/4	15	12
新國民便當-辣味雞腿	2014/10/1	25	25

<參考圖>

2. 設計說明：

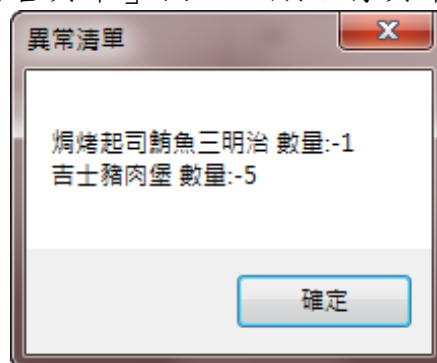
- (1) 請於專案中新增一個 ADO.NET 實體資料模型，將資料庫的資料表及預存程序帶入此實體資料模型。
- (2) 在表單初始化事件中，初始化實體資料模型，並以 Database.Log 的方法寫入到 **log.txt**，可以使用記事本開啟 **log.txt** 檔案，檢視所有執行指令記錄。
- (3) 在 Form1_Load 事件載入 Inventory 表格內容至 list 物件集合。
- (4) 在 tscomboBox 的選取索引變更事件 (SelectedIndexChanged) 中，將 list 物件集合的分類產品資料加入至下拉選單中，當選擇下拉選單的某一項產品分類時，DataGridView 控制項只顯示屬於該分類的產品項目。
- (5) DataGridView 控制項畫面顯示產品、進貨日期、進貨數量、銷售數量等欄位，不能新增及刪除資料，只能編輯進貨數量及銷售數量欄位，當資料有異動時須立即更新到資料庫。
- (6) 資料庫產品欄位的資料目前是 Base64String，在顯示至 DataGridView 控制項時，需透過程式內建的 Decryption 進行轉換至 UTF8 格式。
- (7) 轉換成正常文字顯示透過 CellFormatting 事件來轉換。
- (8) (進貨數量-銷售數量<0)即為異常狀況，已將計算方式寫在 Sales.mdf 資料庫的預存程序 Saleberrant 內，請在工具列上按下「銷售異常」按鈕，執行 Saleberrant 預存程序的資料，顯示有異常銷售的清單資訊。

3. 執行結果參考畫面：

(1) 於下拉選單挑選產品分類，下方產品銷售清單資料依據分類變更顯示。

產品	進貨日期	進貨數量	銷售數量
蛋起湖便當	2014/10/3	10	8
新國民便當-排骨	2014/10/2	20	8
新國民便當-油雞	2014/10/4	15	12
新國民便當-辣味雞腿	2014/10/1	25	25

(2) 在工具列上按下「銷售異常」按鈕，顯示有異常銷售的清單資訊。



(3) 於專案資料夾\bin\Debug 內，以記事本開啟 **log.txt** 檔案，檢視所有執行指令記錄。

```

log.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
執行-- 於 1 毫秒後完成，結果: SqlDataReader
UPDATE [dbo].[Inventory]
SET [銷售數量] = @0
WHERE ([產品] = @1)
-- @0: '6' (Type = Int32)
-- @1: '54SX540k6LW35Y+46a6q6a2a5LiJ5pi05rK7' (Type = String, Size = 50)
-- 於 2014/12/16 下午 04:57:39 +08:00
執行-- 於 0 毫秒後完成，結果: 1

[dbo].[Saleaberrant]
-- 於 2014/12/16 下午 04:57:40 +08:00
執行-- 於 0 毫秒後完成，結果: SqlDataReader

[dbo].[Saleaberrant]
-- 於 2014/12/16 下午 04:58:05 +08:00
執行-- 於 0 毫秒後完成，結果: SqlDataReader

SELECT
    [Extent1].[產品] AS [產品],
    [Extent1].[產品分類] AS [產品分類],
    [Extent1].[進貨日期] AS [進貨日期],
    [Extent1].[進貨數量] AS [進貨數量],
    [Extent1].[銷售數量] AS [銷售數量]
FROM [dbo].[Inventory] AS [Extent1]
-- 於 2014/12/16 下午 04:58:50 +08:00
執行-- 於 12 毫秒後完成，結果: SqlDataReader

[dbo].[Saleaberrant]
-- 於 2014/12/16 下午 04:58:54 +08:00
執行-- 於 1 毫秒後完成，結果: SqlDataReader
  
```