

通識計算機程式設計期中考試題, 4/20/2012

共 8 頁，滿分 100 分

1. 撰寫一或數個 C#敘述達成下列要求: (假設 **using System;** 敘述已經包含於程式中)
 - (a) 宣告 **int** 變數 **k**, **bool** 變數 **b**, **double** 變數 **x** (3%)
 - (b) 在螢幕顯示一行字，要求使用者輸入一個浮點數 (3%)
 - (c) 自鍵盤讀入一個浮點數，並將其值存入已宣告之 **double** 變數 **x** (3%)
 - (d) 將已宣告設值之 **double** 變數 **x** 強制轉型為整數，存入已宣告之 **int** 變數 **k** (3%)
 - (e) 令已宣告之 **bool** 變數 **b** 等於關係式 **k>0** (3%)
2. 撰寫一或數個 C#敘述達成下列要求: (假設 **using System;** 敘述已經包含於程式中)
 - (a) 只用一個敘述，將已宣告設值之 **int** 變數 **a** 設值 (assign) 給他處已宣告之 **int** 變數 **b**，再以遞增算子將 **a** 加 1 (3%)
 - (b) 宣告 **bool** 變數 **r**，並設其值為 **p** 且 (and) (非 **q**) 之邏輯演算的結果。**bool** 變數 **p** 與 **q** 均已於他處宣告設值 (3%)
 - (c) 宣告 **double** 變數 **y**，並令其值為他處已宣告設值之 **double** 變數 **t** 與 2π 之積加上一個他處已宣告設值之 **double** 變數 **phi** 之後的正弦函數 (sine) 值 (數學上即 $y = \sin(2\pi + \text{phi})$)，注意此處 π 為圓周率 (3%)
 - (d) 宣告 **double** 變數 **heaviside**，利用三元運算子使其在他處已宣告設值之 **double** 變數 **x** 大於等於 0 時，**heaviside** 等於 1.0，反之則 **heaviside** 等於 0.0 (3%)
 - (e) 宣告變數 **c** 為 **char** 型別，並令其值為 "換行" (new line) 字元 (3%)
3. 撰寫一或數個 C#敘述達成下列要求: (假設 **using System;** 敘述已經包含於程式中)
 - (a) 宣告一個亂數產生器物件 **rand**，讓系統以時間和網路卡號自動決定種子數 (3%)
 - (b) 宣告一個 **int** 二維陣列
$$\begin{bmatrix} 4 & 9 & 2 \\ 3 & 5 & 7 \\ 8 & 1 & 6 \end{bmatrix}$$
，命名為 **a**，用迴圈計算左上到右下對角線三元素的和，存入 **int** 變數 **sum** (3%)
 - (c) 寫一個無窮迴圈，迴圈中讀入一個整數，讀到一個負整數時跳出迴圈，否則繼續 (3%)
 - (d) 利用 **Console.ReadLine().ToCharArray()** 將讀入之字串轉為字

元陣列 **s**，並用以分設字串第一個字元和最後一個字元為 **char** 變數 **cFirst** 與 **cLast** (3%)

- (e) 寫一個 **void** 函數 **ComputeCubeVolumeAndSurfaceArea**，傳入一個立方體的邊長，傳回此立方體的體積(邊長的立方)和表面積(6 乘以邊長的平方)。再寫一段程式敘述，呼叫此函數，計算邊長為 2 的立方體體積和表面積 (3%)

4. 找出以下程式片段之錯誤，並予更正，使其中的 **Debug.Assert** 敘述(若有的話)成立無誤。

- (a) (3%)

```
int m = 5;
int q = m++;
Debug.Assert(q == 6);
```

- (b) (3%)

```
enum Status
{
    INSIDE,
    OUTSIDE
}
. . . . .
Status s = INSIDE;
```

- (c) (3%) 累加 1 到 100

```
int sum = 0;
int n = 1;
while (n <= 100)
{
    sum += n;
};
Debug.Assert(sum == 5050);
```

- (d) (3%) 呼叫 **void** 函式 **Triple**，使 **a** 變成原值的三倍

```
int a = 3;
Triple(a);
Debug.Assert(a==9);
. . . . .
static void Triple(int a)
```

```

{
    a *= 3;
}

```

(e) (3%) 下列程式片段不應影響陣列 **x**，且更正時不能改變敘述 **y[1]=4**;

```

int[] x = { 2, 1, 2, 5, 3 };
int[] y = x;
y[1] = 4;
Debug.Assert(x[1] == 1);

```

5. 試寫出下列程式的輸出 (5%)

```
using System;
```

```
namespace CPMidTerm2012Problem5
```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            int[] x = { 2, 1, 2, 5, 3 };
            foreach( int element in x )
            {
                Console.Write(element+"\t");
            }
            Console.WriteLine();
            Array.Reverse(x);
            foreach (int element in x)
            {
                Console.Write(element + "\t");
            }
            Console.WriteLine();
            Array.Sort(x);
            foreach (int element in x)
            {
                Console.Write(element + "\t");
            }
            Console.WriteLine();
            int idx = Array.IndexOf(x, 2);

```

```

        Console.WriteLine(idx);
        idx = Array.LastIndexOf(x, 2);
        Console.WriteLine(idx);
    }
}
}

```

6. 試寫出下列程式的輸出 (10%)

```

using System;

namespace CPMidTerm2012Problem6
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] x = { 2, 1, 2, 5, 3 };
            int y = 0;
            const int N = 5;
            int n;
            Console.WriteLine("y[0] = " + y);
            int xCurrent;
            int xPrevious;
            int yPrevious;
            for (n = 1; n < N; ++n)
            {
                xCurrent = x[n];
                xPrevious = x[n - 1];
                yPrevious = y;
                y = IIR(xCurrent, xPrevious, yPrevious);
                Console.WriteLine("y[{0}] = {1}", n, y);
            }
        }

        static int IIR(int xCurrent, int xPrevious,
                       int yPrevious)
        {

```

```

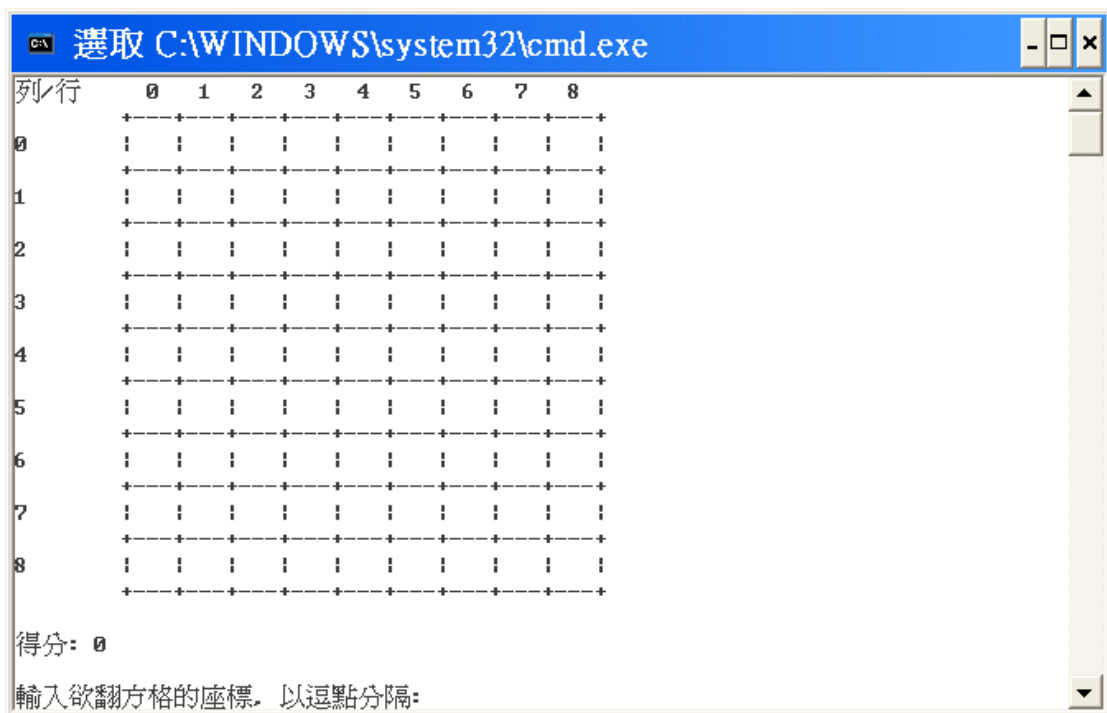
        int y = 2 * xCurrent + xPrevious + yPrevious;
        return y;
    }
}
}

```

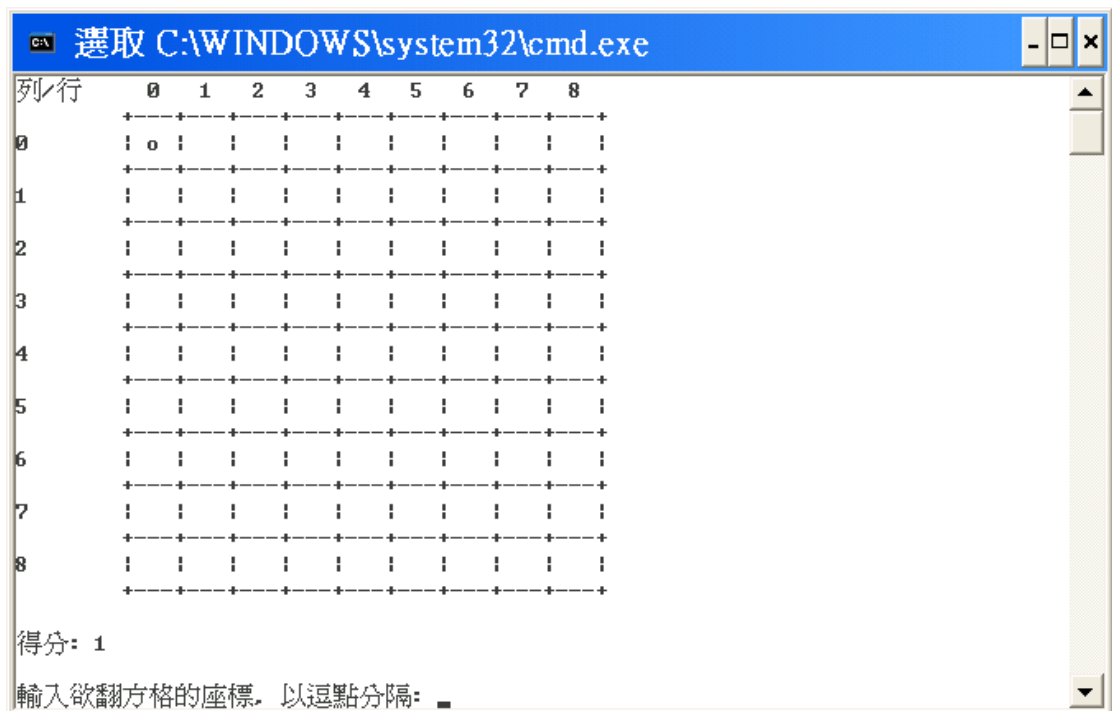
7. 「踩地雷」是微軟 Windows 作業系統所附的小遊戲，許多人都玩過。本題要求寫一個簡化版的踩地雷遊戲，規則如下：

棋盤由 9X9 的方格組成，電腦先隨機於其中 10 個方格佈下地雷，使用者隨後不停任意翻開方格。直到遇見地雷，遊戲即告結束，使用者得分即遇到地雷前翻開的方格總數。

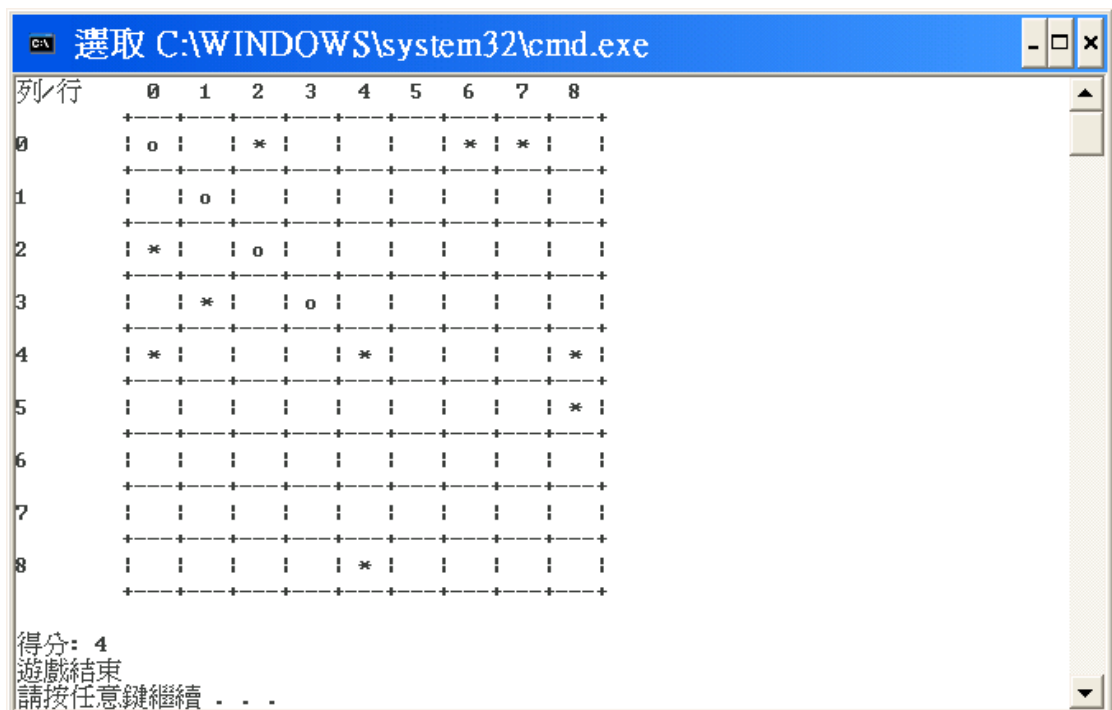
所寫程式的主控台螢幕，開始時如下圖：



使用者輸入座標，代表要翻的方格位置。若沒碰到地雷，清除螢幕，重新顯示棋盤及得分。假設使用者翻開第 0 列第 0 行方格，且未遇地雷，則畫面顯示如下：



圖中可見翻出之方格以圈號表示，並秀出得 1 分。如此反覆進行，直到遇見地雷，遊戲結束，主控台螢幕顯示所有地雷 (以星號表出)及已、未翻方格與得分。一種可能畫面如下：



你可以使用下列已準備好的函式(也可以不用而自行撰寫)：

```

// 顯示棋盤及得分
// 參數說明：
// board代表棋盤
// ix, jx分別代表10顆地雷在棋盤中的列和行座標構成的一維陣列
// 其元素例如ix[5] = 3, jx[5] = 2表示第6顆地雷位在board[3,2]
// score代表使用者得分
// gameOver代表遊戲是否已經結束
static void Display(char[,] board, int[] ix, int[] jx,
                    int score, bool gameOver)
{
    int m;
    int i;
    int j;
    if (gameOver)
    {
        for (m = 0; m < 10; ++m)
        {
            board[ix[m], jx[m]] = '*';
        }
    }
    Console.Write("列/行\t");
    for (j = 0; j < 9; ++j)
    {
        Console.Write(" {0} ", j);
    }
    Console.WriteLine();
    Console.Write("\t");
    for (j = 0; j < 9; ++j)
    {
        Console.Write("+---");
    }
    Console.WriteLine("+");
    for (i = 0; i < 9; ++i)
    {
        Console.Write("{0}\t", i);
        for (j = 0; j < 9; ++j)
        {
            Console.Write("| {0} ", board[i, j]);

```

```

    }
    Console.WriteLine("|");
    Console.Write("\t");
    for (j = 0; j < 9; ++j)
    {
        Console.Write("+---");
    }
    Console.WriteLine("+");
}
Console.WriteLine();
Console.WriteLine("得分: {0}", score);
}

```

上列函式可直接在答案中引用，不需重抄一遍。此處使用陣列為參數，可以先不必考慮傳值或傳址，直接以主程式中的對應陣列作為真實參數呼叫即可。能利用虛擬碼思考，並有適當註解，才可得本題滿分 25 分。(25%)