

通識計算機程式設計期中考參考解答

4/27/2018

1.

(a) (3%)

```
bool isEven;  
int p;  
double s;
```

(b) (3%)

```
Console.WriteLine("輸入一個正整數 p");
```

(c) (3%)

```
p = int.Parse(Console.ReadLine());
```

或

```
p = Convert.ToInt16(Console.ReadLine());
```

(d) (3%)

```
isEven = ((p % 2) == 0);
```

(e) (3%)

```
if(isEven)  
{  
    Console.WriteLine("p為偶數");  
}  
else  
{  
    Console.WriteLine("p為奇數");  
}
```

2.

(a) (3%)

```
m = ++n;
```

(b) (3%)

```
r = n % m;
```

(c) (3%)

```
double y = Math.Sin(2.0 * Math.PI / lambda * x);
```

(d) (3%)

```
y = (x > 0.0) ? 1.0 : 0.0;
```

(e) (3%)

```
char c = '\\t';
```

3.

(a) (3%)

```
Random rand = new Random();
```

(b) (3%)

```
double[] x = { 0.2, 0.3, 0.1 };
```

```
double[] w = { 0.5, 0.1, 0.2 };
```

(c) (3 %)

```
int n = x.Length;
```

或

```
int n = x.GetUpperBound(0) + 1;
```

(d) (3%)

```
double y = 0.2;
for(int i = 0; i < n; ++i)
{
    if(rand.Next() % 2 == 0)
    {
        y += w[i] * x[i];
    }
}
double output = Sigmoid(y);
```

(e) (3%)

```
static double Sigmoid(double y)
{
    double output = 1.0 / (1.0 + Math.Exp(y));
}
```

```

        return output;
    }

```

4. 找出以下程式片段之錯誤，並在盡量保持原先程式碼之前提下，予以更正。

假設 `using System;` 敘述已經包含於程式中。

(a) (3%) (一個語法錯誤)

控制變數 `n` 宣告為 `const` 後，其值不可改變。

將 `const` 關鍵字刪除，即可得到所欲輸出。

```

const int n = 1;
int sum = 0;
while (n <= 100)
{
    sum += n;
    ++n;
}
Console.WriteLine("sum = {0}", sum);

```

(b) (3%) (一個語義錯誤)

此為懸置 `if` 問題。

「`else z = c / b;`」對應的是「`if (b == 0) z = -1;`」，

而非所欲之「`if (a == 0)`」。

使用大括弧分別清楚，便可得到所要的螢幕輸出。

```

int a = 2;
int b = 2;
int c = -4;
int z = 0;
if (a == 0)
{
    if (b == 0) z = -1;
}
else
{
    z = c / b;
}

```

```
Console.WriteLine("z = {0}", z);
```

(c) (3%) (一個語法錯誤)

取用 **enum** 常數，需在前註明 **enum** 型別名稱。

```
enum Diagnosis
{
    NORMAL,
    MCI
}
. . . . .
Diagnosis diagnosis = Diagnosis.MCI;
switch(diagnosis)
{
    case Diagnosis.NORMAL:
        Console.WriteLine("Normal");
        break;
    case Diagnosis.MCI:
        Console.WriteLine("MCI");
        break;
    default:
        Console.WriteLine("Invalid diagnosis");
        break;
}
```

(d) (3%) (一個語義錯誤)

當 **x** 累積到 **y** 值，即跳離迴圈；以致無法顯示最後一個因數。

將跳離迴圈的條件，改成 **x** 大於 **y** 才能跳出，即可在螢幕顯示 **y** 本身數值，亦即 **y** 的最大因數。

```
int x = 2;
int y = 12;
for(; ;)
{
    if(y % x == 0)
    {
        Console.WriteLine(x);
    }
    ++x;
}
```

```

        if (x >= y) break;
    }

```

(e) (3%) (一個語義錯誤)

使用傳值呼叫，將 **matrix** 的參考複製一份，傳入函式 **InverseAndReset**。所以函式 **InverseAndReset** 內，改變的 **matrix** 參考，不會影響到主程式中的 **matrix** 參考，無法將之歸零。

在此只要採用傳址呼叫，使 **InverseAndReset** 中對於陣列 **matrix** 的改變，也同時改變主程式中的 **matrix** 參考，便可完成所需的功能。不論傳值或傳址呼叫，二維陣列 **copy** 均不受影響，因此能維持原先設定的值。

注意函式內更改輸入陣列的參考，可能產生程式設計師意料之外的結果，不是好的程式寫法，請不要模仿。

```

static void Main(string[] args)
{
    int[,] matrix = { {1, 1},
                      {3, 4} };
    int[,] copy = matrix;
    InverseAndReset(ref matrix);
    Console.WriteLine("Original matrix = ");
    Console.WriteLine(
        "{0} \t {1}", copy[0,0], copy[0,1]);
    Console.WriteLine(
        "{0} \t {1}", copy[1,0], copy[1,1]);
    Console.WriteLine("Reset matrix = ");
    Console.WriteLine(
        "{0} \t {1}", matrix[0, 0], matrix[0, 1]);
    Console.WriteLine(
        "{0} \t {1}", matrix[1, 0], matrix[1, 1]);
}

static void InverseAndReset(ref int[,] matrix)
{
    // inverse
    int[,] inverse = new int[2,2];
    int delta =
        matrix[0, 0] * matrix[1, 1] -

```

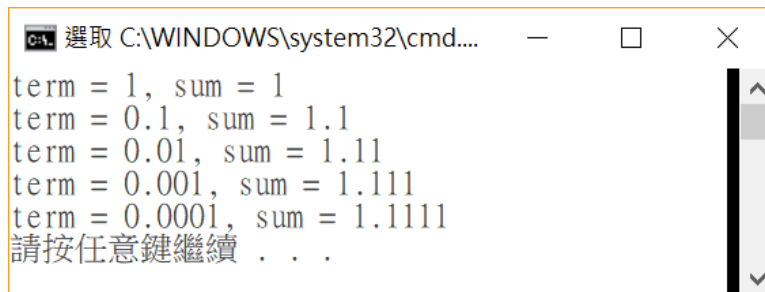
```

        matrix[0, 1] * matrix[1, 0];
inverse[0, 0] = matrix[1, 1]/delta;
inverse[1, 1] = matrix[0, 0]/delta;
inverse[0, 1] = matrix[1, 0]/delta;
inverse[1, 0] = matrix[0, 1]/delta;
matrix = inverse;

// reset
matrix = new int[2, 2];
matrix[0, 0] = 0;
matrix[0, 1] = 0;
matrix[1, 0] = 0;
matrix[1, 1] = 0;
}

```

5. (5 %)

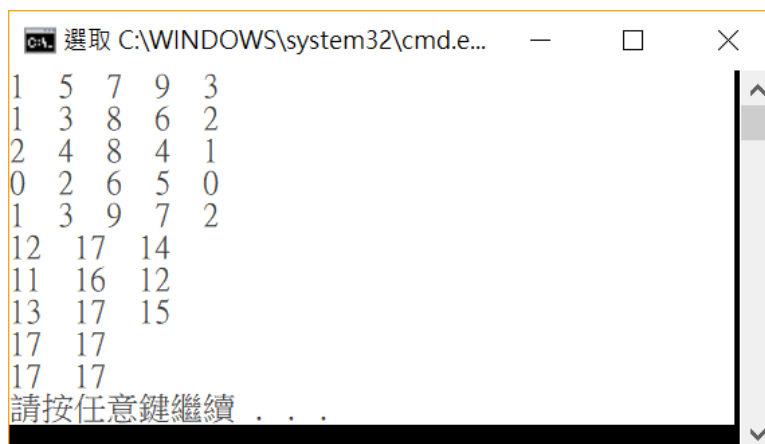


```

C:\A 選取 C:\WINDOWS\system32\cmd...
term = 1, sum = 1
term = 0.1, sum = 1.1
term = 0.01, sum = 1.11
term = 0.001, sum = 1.111
term = 0.0001, sum = 1.1111
請按任意鍵繼續 . . .

```

6. (10 %)



```

C:\A 選取 C:\WINDOWS\system32\cmd.e...
1 5 7 9 3
1 3 8 6 2
2 4 8 4 1
0 2 6 5 0
1 3 9 7 2
12 17 14
11 16 12
13 17 15
17 17
17 17
請按任意鍵繼續 . . .

```

7.

// Problem 7

```

// skj 4/23/2018
//
// 計算語料庫中各受試者說話
// 句中的名詞(N), 代名詞(P), 動詞(V)發生率
// 名詞發生率  $NR = N/W$ 
// 代名詞發生率  $PR = P/W$ 
// 動詞發生率  $VR = V/W$ 
// 其中 W 代表全句中的字詞總數
//
// 語料庫結構(二維陣列 Corpus)
// M 個受試者, 每人不超過 Q 個字詞 (M 列, Q行)
// 每個受試者另加一個subject name(受試者代號)
// 以及狀態標示(Normal或MCI)
//
// 語料庫範例(4個受試者, 最多7個字詞, 字詞不足以'x'取代)
// 受試者A MCI 他 站 在 椅子 上 拿 東西
// 受試者B Normal 男孩 站 在 椅子 上 x x
// 受試者C MCI 樓上 拿 東西 到 樓上 拿 東西
// 受試者D Normal 男孩 到 櫥櫃 上 拿 餅乾 x
//
// 詞典結構 (一維陣列)
// 由同詞性的詞構成構成
// nouns, pronouns, verbs
//
// 虛擬碼
// 1. 建立語料庫及詞典
// 2. for each row of corpus (每一受試者)
// 2.1 累計W
// 2.2 計算 $NR = N/W$ 
// 2.3 計算 $PR = P/W$ 
// 2.4 計算 $VR = V/W$ 
// 2.5 輸出 受試者代碼, MCI或Normal, W, NR, PR, VR
//

using System;
namespace Problem7
{

```

```

public class Program
{
    static void Main(string[] args)
    {
        string[,] corpus = {
            {"受試者A", "MCI",
            "他", "站", "在", "椅子", "上", "拿", "東西", "這",
            "水龍頭", "放", "的", "水", "在", "洗", "東西", "洗",
            "窗戶", "洗", "碗", "他", "站", "在", "椅子", "上",
            "拿", "東西", "凳子", "水龍頭"},
            {"受試者B", "Normal",
            "男孩", "站", "在", "椅子", "上", "拿", "東西", "媽媽",
            "用", "水龍頭", "的", "水", "洗", "碗", "X", "X",
            "X", "X", "X", "X", "X", "X", "X", "X",
            "X", "X", "X", "X"},
            {"受試者C", "MCI",
            "他", "站", "在", "椅子", "上", "拿", "東西", "這",
            "水龍頭", "放", "的", "水", "在", "洗", "東西", "洗",
            "窗戶", "洗", "碗", "他", "站", "在", "椅子", "上",
            "拿", "東西", "凳子", "水龍頭"},
            {"受試者D", "Normal",
            "男孩", "到", "櫥櫃", "上", "拿", "餅乾", "婦女", "在",
            "水池", "裡面", "洗", "碗", "X", "X", "X", "X",
            "X", "X", "X", "X", "X", "X", "X", "X", "X", "X", "X", "X"}
        };

        string[] nouns = {
            "椅子", "東西", "水龍頭", "水", "窗戶", "碗", "凳子", "男孩",
            "媽媽", "櫥櫃", "餅乾", "婦女", "水池", "裡面"};

        string[] pronouns = {"他", "這"};
        string[] verbs = {"站", "拿", "放", "洗", "用"};

        //*****
        for(int subjectID = 0;
            subjectID <= corpus.GetUpperBound(0);
            ++subjectID)
        {

```



```

        int w = SizeOfVocabulary(subjectID, corpus);
        double nNR = RateOfPartOfSpeech(
            subjectID, corpus, nouns, w);
        double pPR = RateOfPartOfSpeech(
            subjectID, corpus, pronouns, w);
        double vVR = RateOfPartOfSpeech(
            subjectID, corpus, verbs, w);
        DisplayFeatures(
            subjectID,
            corpus[subjectID, 0], corpus[subjectID, 1],
            nNR, pPR, vVR);
    }
    //*****
}

public static int SizeOfVocabulary(
    int subjectID, string[,] corpus)
{
    int w = 0;
    for(int i = 2; i <= corpus.GetUpperBound(1); ++i)
    {
        if (corpus[subjectID, i] == "X") break;
        ++w;
    }
    return w;
}

public static double RateOfPartOfSpeech(
    int subjectID,
    string[,] corpus, string[] dictionary, int w)
{
    string word;
    int nPartOfSpeech = 0;
    for(
        int i = 2; i <= corpus.GetUpperBound(1); ++i)
    {
        word = corpus[subjectID, i];
        if (word == "X") break;

```

```

        if (Array.IndexOf(dictionary, word) < 0)
            continue;
        ++nPartOfSpeech;
    }
    double ratio =
        (double) nPartOfSpeech / (double) w;
    return ratio;
}

public static void DisplayFeatures(
    int subjectID,
    string name, string diagnos,
    double nNR, double pPR, double vVR)
{
    Console.WriteLine(
        "{0}\t {1}\t {2}\t NR = {3:F6}\t PR = {4:F6}\t VR = {5:F6}",
        subjectID, name, diagnosis, nNR, pPR, vVR);
}
}
}

```

本題滿分 25 分，全部程式集中寫成一個大 **Main** 函式，不區分函式者，最高得 20 分；善用函式，乃至尚未教到的物件導向程式設計(object-oriented programming)者，最高得 23 分；能利用虛擬碼或流程圖思考，適當劃分函式或類別(class)者，最高得 25 分(使用虛擬碼)或 24 分(使用流程圖)。(25%)