

通識計算機程式設計期中考參考解答

臺灣大學 鄭士康 4/24/2015



本講義除另有註明外，採創用CC姓名標示-非商業性-相同方式分享3.0臺灣版授權釋出

1.

- (a) (3%)
`bool bl;`
`int m;`
`byte b;`
- (b) (3%)
`Console.WriteLine("輸入一個整數");`
- (c) (3%)
`m = int.Parse(Console.ReadLine());`
- (d) (3%)
`b = (byte) m;`
- (e) (3%)
`bl = (m >= 0 && m < 256);`

2.

- (a) (3%)
`int n = m--;`
- (b) (3%)
`int q = n / 5;`
- (c) (3%)
`double db = 20.0 * Math.Log10(Math.Abs(x));`
- (d) (3%)
`int sgn = (u < 0.0) ? -1 : 1;`
- (e) (3%)
`char c = '\n';`

3.

- (a) (3%)
`Random rand = new Random(777);`
`int weekDay = rand.Next() % 7;`
- (b) (3%)
`const int ARRAY_SIZE = 5;`
- (c) (3%)
`int workHours;`
`do`
`{`
`workHours = int.Parse(Console.ReadLine());`
`} while (workHours > 24 || workHours < 0);`
- (d) (3%)
`char[] characters =`
`Console.ReadLine().ToCharArray();`
`Array.Sort(characters);`
`Array.Reverse(characters);`
`char largest = characters[0];`

(e) (3%) `static void Initialize(out double x)`
 `{`
 `x = 1.0;`
 `}`

4.

(a) (3%) 以下紅字標示的 '+' 號意義混淆：可看成字串合併，也可解釋成算術加號；須以圓括弧表示先加1後，再自動將結果整數值轉換為字串，與前面字串合併。

```
int i = 0;  
// (誤) Console.WriteLine("i + 1 = " + i + 1);  
Console.WriteLine("i + 1 = " + (i + 1));     // (正)
```

(b) (3%) 整數 `n` / 整數 `2` 代表其商，沒有小數部分；隨後再設值為 `double` 變數時，已無小數部分。應先將 `n` 或 `2` 強制轉型為 `double`，即可進行 `double` 除以 `double` 之計算，得到小數部分。

```
int n = 1;  
// (誤) double fraction = n / 2;  
double fraction = (double) n / 2;     // (正)  
Console.WriteLine(fraction);
```

(c) (3%) 此迴圈開始計算 `sum` 時的 `i` 值為 `2`，不是所要的 `1`，因此須將 `i` 宣告時所設初值 `1` 減為 `0`。

```
// (誤) int i = 1;  
int i = 0;     // (正)  
int sum = 0;  
while(i < 100)  
{  
    ++i;  
    sum += i;  
}  
Console.WriteLine("1 + 2 + 3 + . . . + 100 = " + sum);
```

(d) (3%) 函式中亂數產生器以被呼叫時的時間及網路卡產生一個亂數種子。本題 `j` 和 `k` 先後連續呼叫 `ARandomNumber`，對應時間差小於計算機計

時的解析度，因此產生的亂數種子相同，接下來傳回的亂數也會相同。此問題解決方法之一便是讓呼叫 **ARandomNumber** 的時間差距夠大，亦即先以 **Thread.Sleep** 延遲足夠時間。

```
using System.Threading; // (正, 插入之敘述)
. . .
int k = ARandomNumber();
Thread.Sleep(1000); // (正, 插入之敘述)
int j = ARandomNumber();
Console.WriteLine("k = {0}, j = {1}", k, j);

. . . . .
static int ARandomNumber()
{
    Random rand = new Random();
    return rand.Next();
}
```

- (e) (3%) 原先將陣列**a**直接設值給**b**時，將**a**陣列在**Heap**記憶區所佔空間的地址參考(reference)，複製給**b**陣列；因此**a**和**b**對應的是同一塊記憶區，等於是同一個陣列。當**b**陣列更動時，**a**陣列也跟著變動。若要使**a**陣列保持原內容，便不能將**a**參考直接設值給**b**，而只能將**a**內容利用**Array.Copy**或其他方式複製給**b**。

```
int[] a = { 1, 2, 3 };
// (誤) int[] b = a;
int[] b = new int[3]; // (正)
Array.Copy(a, b, a.Length); // (正, 插入之敘述)
b[0] = 7;
b[1] = 8;
for(int i = 0; i < a.Length; i++)
{
    Console.Write(a[i] + "\t");
}
Console.WriteLine();
for(int i = 0; i < b.Length; i++)
{
    Console.Write(b[i] + "\t");
}
```

```
}
Console.WriteLine();
```

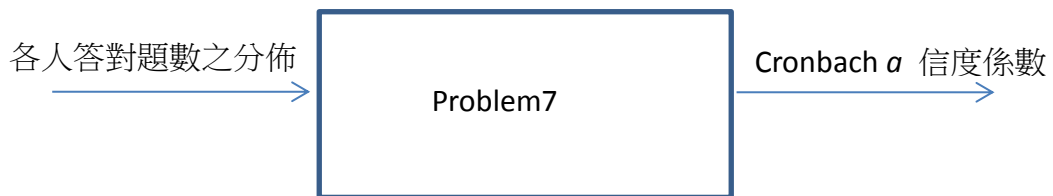
5. (5 %) 螢幕輸出如下：

```
C:\>
0      0
1      1
2      0
3      1
4      1
Total number of correct choices = :3
請按任意鍵繼續 . . .
```

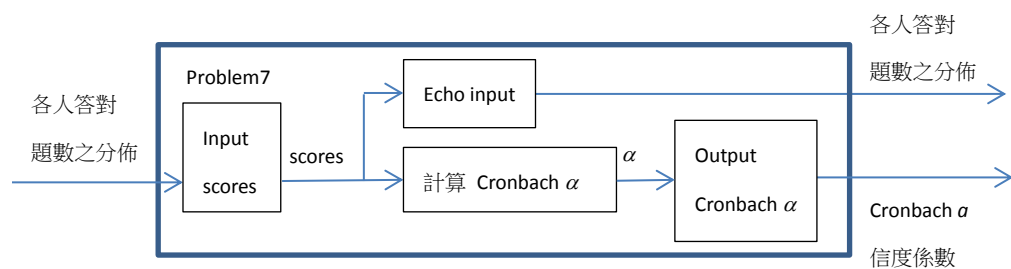
6. (10 %) 螢幕輸出如下：

```
C:\>
1      2      1      3      1      4      2
2      1      2      4      1      3      1
3      1      2      1      3      4      1
4      1      2      1      3      1
請按任意鍵繼續 . . .
```

7. 程式黑箱系統方塊圖:



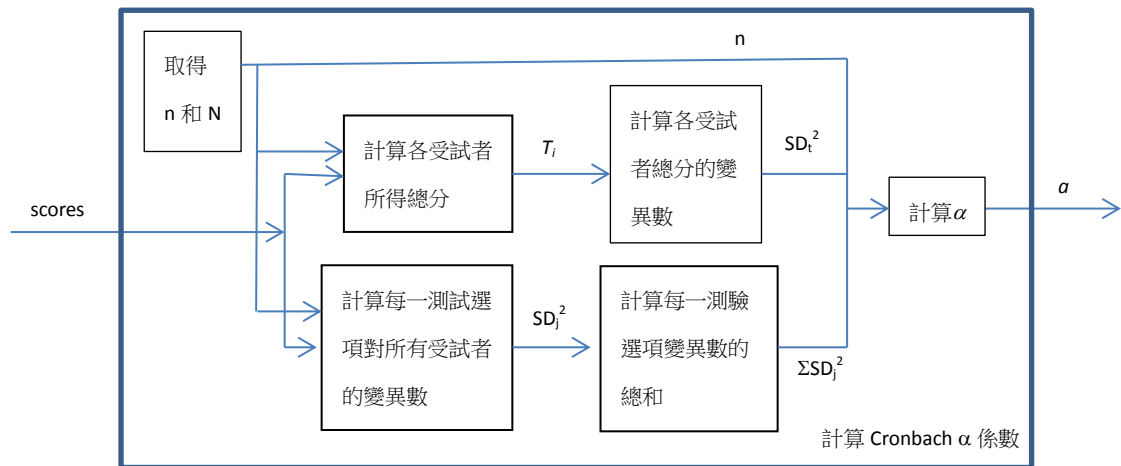
主程式系統分解(灰箱)方塊圖:



虛擬碼: 主程式

1. Input scores
2. Echo input
3. 計算 Cronbach α 係數
4. Output Cronbach α

計算 Cronbach α 係數系統分解方塊圖:



虛擬碼: 計算 Cronbach α 係數(二維陣列 scores)

1. 由 scores 取得 n 和 N
2. 計算各受試者所得總分，得到元素為 T_i 的一維陣列
3. 計算各受試者總分的變異數 $SD_t^2 = \frac{\sum (T_i - \bar{T})^2}{N}$
4. 計算每一測驗選項對所有受試者的變異數 $SD_j^2 = \frac{\sum (s_{ij} - \bar{s}_j)^2}{N}$
5. 計算每一測驗選項變異數的總和 $\sum SD_j^2$
6. 計算 $\alpha = \frac{n}{n-1} \left(1 - \frac{\sum SD_j^2}{SD_t^2} \right)$

單元測試專案 UnitTestProject1

```

using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Problem7;

namespace UnitTestProject1
{
    [TestClass]
    public class UnitTest1
    
```

```

{
    [TestMethod]
    public void TestSumOfColumns()
    {
        int[,] scores = {
                                {6, 8, 9, 1, 7},
                                {8, 8, 8, 2, 4},
                                {0, 3, 6, 1, 3},
                                {2, 1, 1, 0, 2}
                            };

        double[] expected = { 31, 30, 13, 6 };
        double[] actual = Program.SumOfColumns(scores);
        Assert.IsTrue(
            MeanSquareError(actual, expected) < 1.0e-8);
    }

    [TestMethod]
    public void TestAverage()
    {
        double[] totals = {31, 30, 13, 6};
        double expected = 20;
        double actual = Program.Average(totals);
        Assert.IsTrue(
            Math.Abs(actual - expected) < 1.0e-8);
    }

    [TestMethod]
    public void TestVariance()
    {
        double[] totals = {31, 30, 13, 6};
        double expected = 116.5;
        double actual = Program.Variance(totals);
        Assert.IsTrue(
            Math.Abs(actual - expected) < 1.0e-7);
    }

    [TestMethod]
    public void TestVariancesOfColumns()

```

```

{
    int[,] scores = {
        {6, 8, 9, 1, 7},
        {8, 8, 8, 2, 4},
        {0, 3, 6, 1, 3},
        {2, 1, 1, 0, 2}
    };
    double[] expected = { 10.0, 9.5, 9.5, 0.5, 3.5 };
    double[] actual =
        Program.VariancesOfColumns(scores);
    Assert.IsTrue(
        MeanSquareError(actual, expected) < 1.0e-8);
}

```

```

[TestMethod]
public void TestSumOfElements()
{
    double[] variances = { 10.0, 9.5, 9.5, 0.5, 3.5 };
    double expected = 33.0;
    double actual = Program.SumOfElements(variances);
    Assert.IsTrue(
        Math.Abs(actual - expected) < 1.0e-7);
}

```

```

[TestMethod]
public void TestCronbachAlpha()
{
    int[,] scores = {
        {6, 8, 9, 1, 7},
        {8, 8, 8, 2, 4},
        {0, 3, 6, 1, 3},
        {2, 1, 1, 0, 2}
    };
    double expected = 0.9;
    double actual = Program.CronbachAlpha(scores);
    Assert.IsTrue(
        Math.Abs(actual - expected) < 1.0e-2);
}

```

```

public double MeanSquareError(
    double[] actual, double[] expected)
{
    int n = actual.Length;
    double sumOfSquareErrors = 0.0;
    for (int i = 0; i < n; i++)
    {
        double error = actual[i] - expected[i];
        sumOfSquareErrors += error * error;
    }
    return sumOfSquareErrors / n;
}
}
}

```

程式專案 Problem7

```

using System;
using System.Diagnostics;

namespace Problem7
{
    public class Program
    {
        static void Main(string[] args)
        {
            Debug.Assert(TestScenario_1_OK());

            int[,] scores = InputScores();
            Echo(scores);
            double alpha = CronbachAlpha(scores);
            OutputAlpha(alpha);
        }

        public static double CronbachAlpha(int[,] scores)
        {

```



```

        int n = scores.GetUpperBound(1) + 1;
        double[] total = SumOfColumns(scores);
        double variance_t = Variance(total);
        double[] variances = VariancesOfColumns(scores);
        double sumOfVariances = SumOfElements(variances);
        double alpha =
(double)n / (n - 1) * (1.0 - sumOfVariances / variance_t);
        return 0.9;
    }

```

```

public static double[] SumOfColumns(int[,] scores)
{
    int n = scores.GetUpperBound(0) + 1;
    double[] totals = new double[n];
    for(int i = 0; i < n; i++)
    {
        totals[i] = 0.0;
        for(int j = 0;
            j < scores.GetUpperBound(1)+1; j++)
        {
            totals[i] += scores[i, j];
        }
    }
    return totals;
}

```

```

public static double Variance(double[] totals)
{
    double mean = Average(totals);
    double n = totals.Length;
    double sumOfSquareError = 0.0;
    for(int i = 0; i < n; i++)
    {
        double error = totals[i] - mean;
        sumOfSquareError += error * error;
    }
    return sumOfSquareError / n;
}

```

```

public static double Average(double[] totals)
{
    int n = totals.Length;
    double sum = 0.0;
    for(int i = 0; i < n; i++)
    {
        sum += totals[i];
    }
    return sum / n;
}

```

```

public static double[] VariancesOfColumns(
    int[, ] scores)
{
    int n = scores.GetUpperBound(0) + 1;
    double[] variances = new double[n];
    double[] data = new double[n];
    for (int j = 0; j < n; ++j)
    {
        for (int i = 0; i < n; i++)
        {
            data[i] = scores[i, j];
        }
        variances[j] = Variance(data);
    }
    return variances;
}

```

```

public static double SumOfElements(
    double[] variances)
{
    double sum = 0.0;
    for(int i = 0; i < variances.Length; i++)
    {
        sum += variances[i];
    }
    return sum;
}

```

```

    }

    public static double MeanSquareError(
        double[] actual, double[] expected)
    {
        int n = actual.Length;
        double sumOfSquareErrors = 0.0;
        for (int i = 0; i < n; i++)
        {
            double error = actual[i] - expected[i];
            sumOfSquareErrors += error * error;
        }
        return sumOfSquareErrors / n;
    }

    // Debug.Assert 測試函式
    public static bool TestScenario_1_OK()
    {
        CronbachAlpha_OK();
        SumOfColumns_OK();
        Average_OK();
        Variance_OK();
        VariancesOfColumns_OK();
        SumOfElements_OK();
        return true;
    }

    public static void CronbachAlpha_OK()
    {
        int[,] scores = {
            {6, 8, 9, 1, 7},
            {8, 8, 8, 2, 4},
            {0, 3, 6, 1, 3},
            {2, 1, 1, 0, 2}
        };

        double expected = 0.9;
        double actual = CronbachAlpha(scores);
        Debug.Assert(

```

```

        Math.Abs(actual - expected) < 1.0e-2);
    }

    public static void SumOfColumns_OK()
    {
        int[,] scores = {
            {6, 8, 9, 1, 7},
            {8, 8, 8, 2, 4},
            {0, 3, 6, 1, 3},
            {2, 1, 1, 0, 2}
        };

        double[] expected = { 31, 30, 13, 6 };
        double[] actual = SumOfColumns(scores);
        Debug.Assert(
            MeanSquareError(actual, expected) < 1.0e-8);
    }

    public static void Average_OK()
    {
        double[] totals = { 31, 30, 13, 6 };
        double expected = 20;
        double actual = Average(totals);
        Debug.Assert(
            Math.Abs(actual - expected) < 1.0e-8);
    }

    public static void Variance_OK()
    {
        double[] totals = { 31, 30, 13, 6 };
        double expected = 116.5;
        double actual = Variance(totals);
        Debug.Assert(
            Math.Abs(actual - expected) < 1.0e-7);
    }

    public static void VariancesOfColumns_OK()
    {
        int[,] scores = {

```

```

        {6, 8, 9, 1, 7},
        {8, 8, 8, 2, 4},
        {0, 3, 6, 1, 3},
        {2, 1, 1, 0, 2}
    };

    double[] expected = { 10.0, 9.5, 9.5, 0.5, 3.5 };
    double[] actual = VariancesOfColumns(scores);
    Debug.Assert(
        MeanSquareError(actual, expected) < 1.0e-8);
}

public static void SumOfElements_OK()
{
    double[] variances = { 10.0, 9.5, 9.5, 0.5, 3.5 };
    double expected = 33.0;
    double actual = SumOfElements(variances);
    Debug.Assert(
        Math.Abs(actual - expected) < 1.0e-7);
}
}
}

```

測試總管[全部執行]後所得到的畫面

