

第5次練習-練習-PC5

學號：109111121

姓名：邱致謙

作業撰寫時間：60 (mins，包含程式撰寫時間)

最後撰寫文件日期：2022/10/03

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

■說明內容

■個人認為完成作業須具備觀念

說明程式與內容

```
int[,] i_Num = new int[3, 3]
{
    {7,8,9 },
    {1,4,3 },
    {2,5,6 }
};
```

```
int[] i_CtNum = new int[19] {
    10000,36,720,360,80,
    252,108,72,54,180,
    72,180,119,36,306,
    1080,144,1800,3600
};
```

以上程式碼是宣告一個二維陣列i_Num和一個一維陣列i_CtNum

```
Response.Write("可以獲得最多的錢是"+mt_GetMost(i_Num, i_CtNum));
Response.Write("<br>");
Response.Write("可以獲得最多的錢是" + mt_GetLeast(i_Num, i_CtNum));
```

以上程式碼是印出最大值和最小值(中間空一行)

```
int mt_GetMost(int[,] i_Num, int[] i_CtNum)
{
    int i_Max = 0;

    for (int i_Row = 0; i_Row < i_Num.GetLength(0); i_Row++)
```

```
{
    int i_Rsum = 0;
    int i_Csum = 0;
    for (int i_Col = 0; i_Col < i_Num.GetLength(1); i_Col++)
    {
        i_Rsum += i_Num[i_Row, i_Col];
        i_Csum += i_Num[i_Col, i_Row];
    }
    // Response.Write((i_Rsum, i_Csum).ToString()); Test

    if (i_CtNum[i_Rsum - 6] > i_CtNum[i_Csum - 6])
    {
        if (i_CtNum[i_Rsum - 6] > i_Max)
        {
            i_Max = i_CtNum[i_Rsum - 6];
        }
    }
    else
    {
        if (i_CtNum[i_Csum - 6] > i_Max)
        {
            i_Max = i_CtNum[i_Csum - 6];
        }
    }
}

int i_Slash1 = i_Num[0, 0] + i_Num[1, 1] + i_Num[2, 2];
if (i_CtNum[i_Slash1 - 6] > i_Max)
{
    i_Max = i_CtNum[i_Slash1 - 6];
}
//Response.Write((i_Slash1).ToString()); Test

int i_Slash2 = i_Num[0, 2] + i_Num[1, 1] + i_Num[2, 0];
if (i_CtNum[i_Slash2 - 6] > i_Max)
{
    i_Max = i_CtNum[i_Slash2 - 6];
}
//Response.Write((i_Slash2).ToString()); Test
return i_Max;
}
```

以上程式碼是我宣告一個i_Max為最大值的變數，用for迴圈一行一列判斷大小，如果比i_Max大，i_Max就更新，最後再判斷兩條斜線的值有沒有大於i_Max，如果比i_Max大，i_Max就更新。

```
int mt_GetLeast(int[,] i_Num, int[] i_CtNum)
{
    int i_Min = 10001;

    for (int i_Row = 0; i_Row < i_Num.GetLength(0); i_Row++)
    {
        int i_Rsum = 0;
        int i_Csum = 0;
        for (int i_Col = 0; i_Col < i_Num.GetLength(1); i_Col++)
        {
            i_Rsum += i_Num[i_Row, i_Col];
            i_Csum += i_Num[i_Col, i_Row];
        }
        //Response.Write((i_Rsum, i_Csum).ToString()); Test

        if (i_CtNum[i_Rsum - 6] < i_CtNum[i_Csum - 6])
        {
            if (i_CtNum[i_Rsum - 6] < i_Min)
            {
                i_Min = i_CtNum[i_Rsum - 6];
            }
        }
        else
        {
            if (i_CtNum[i_Csum - 6] < i_Min)
            {
                i_Min = i_CtNum[i_Csum - 6];
            }
        }
    }

    int i_Slash1 = i_Num[0, 0] + i_Num[1, 1] + i_Num[2, 2];
    if (i_CtNum[i_Slash1 - 6] < i_Min)
    {
        i_Min = i_CtNum[i_Slash1 - 6];
    }
    //Response.Write((i_Slash1).ToString()); Test

    int i_Slash2 = i_Num[0, 2] + i_Num[1, 1] + i_Num[2, 0];
    if (i_CtNum[i_Slash2 - 6] < i_Min)
```

```
{  
    i_Min = i_CtNum[i_Slash2 - 6];  
}  
//Response.Write((i_Slash2).ToString()); Test  
return i_Min;  
}
```

以上程式碼是我宣告一個i_Min為最小值的變數，用for迴圈一行一列判斷大小，如果比i_Min小，i_Min就更新，最後再判斷兩條斜線的值有沒有小於i_Min，如果比i_Min小，i_Min就更新。

個人認為完成作業須具備觀念

這次也是邏輯題目，受到同學啟發，於是我用一行一列的方式來判斷，發現第一行跟第一列的index剛好是相反的，這樣寫起來就輕鬆多了，最後再判斷兩條斜線，不過途中還是遇到很多語法上的問題(譬如int轉型成string)，也反覆的Response測試結果，終於解決。