

第5次練習-練習-PC5

學號：109111101

姓名：邱韋翔

作業撰寫時間：20 (mins，包含程式撰寫時間)

最後撰寫文件日期：2022/10/05

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

先用陣列 `ia_Money` 來儲存所有的金額，在使用一個二為陣列 `ia_2DArr` 代替彩券上3 X 3的正方形格子，接著建立兩個方法，一個用來處理最多錢的和另個則處理最小金錢的和，兩種方法的架構都差不多，首先，把行相加的進行逐一比較，再來是列，最後則是斜線比較，再根據最後是取最大或最小的金錢進行比對，最後回傳值，即可得到最後的結果，下段程式碼為使用後結果:

```
namespace _111_1PC5
{
    public partial class Test : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            int[] ia_Money = new int[19]
            {
                10000, 36, 720, 360, 80,
                252, 108, 72, 54, 180,
                72, 180, 119, 36, 306,
                1080, 144, 1800, 3600
            };

            int[,] ia_2DArr = new int[3, 3]
            {
                { 7, 8, 9 },
                { 1, 4, 3 },
                { 2, 5, 6 },
            };
        }
    }
}
```

```
Response.Write("可獲得最多錢的和為:"+mt_GetMost(ia_2DArr, ia_Money)+"<
Response.Write("可獲得最少錢的和為:"+mt_GetLeast(ia_2DArr, ia_Money));
}
int mt_GetMost(int[,] ia_2DArr, int[] ia_Money)
{
    int i_MaxMoney = 0;
    int i_MaxSum = 0;
    for (int i_Row = 0; i_Row < ia_2DArr.GetLength(0); i_Row++)
    {
        //V 3 Col
        int i_Sum = 0;
        int i_TmpMoney = 0;
        for (int i_Col = 0; i_Col < ia_2DArr.GetLength(1); i_Col++)
        {
            i_Sum += ia_2DArr[i_Row, i_Col];
        }
        i_TmpMoney = ia_Money[(i_Sum - 6)];
        if (i_TmpMoney > i_MaxMoney) {
            i_MaxSum = i_Sum;
            i_MaxMoney = i_TmpMoney;
        }
        i_Sum = 0;
        //V 3 Row
        for (int i_Col = 0; i_Col < ia_2DArr.GetLength(1); i_Col++)
        {
            i_Sum += ia_2DArr[i_Col, i_Row];
        }

        i_TmpMoney = ia_Money[(i_Sum - 6)];
        if (i_TmpMoney > i_MaxMoney)
        {
            i_MaxSum = i_Sum;
            i_MaxMoney = i_TmpMoney;
        }
        //V 2 incline
        i_Sum = ia_2DArr[0, 0] + ia_2DArr[1, 1] + ia_2DArr[2, 2];
        i_TmpMoney = ia_Money[(i_Sum - 6)];
        if (i_TmpMoney > i_MaxMoney)
        {
            i_MaxSum = i_Sum;
            i_MaxMoney = i_TmpMoney;
        }
        i_Sum = ia_2DArr[0, 2] + ia_2DArr[1, 1] + ia_2DArr[2, 0];
        i_TmpMoney = ia_Money[(i_Sum - 6)];
        if (i_TmpMoney > i_MaxMoney)
        {
            i_MaxSum = i_Sum;
```



```
        i_MaxMoney = i_TmpMoney;
    }
}
return i_MaxSum;
}
int mt_GetLeast(int[,] ia_2DArr, int[] ia_Money)
{
    int i_MinMoney = 0;
    int i_MinSum = 0;
    for (int i_Row = 0; i_Row < ia_2DArr.GetLength(0); i_Row++)
    {
        //V 3 Col
        int i_Sum = 0;
        int i_TmpMoney = 0;
        for (int i_Col = 0; i_Col < ia_2DArr.GetLength(1); i_Col++)
        {
            i_Sum += ia_2DArr[i_Row, i_Col];
        }

        i_TmpMoney = ia_Money[(i_Sum - 6)];
        if (i_MinMoney == 0)
        {
            i_MinMoney = i_TmpMoney;
        }
        if (i_TmpMoney <= i_MinMoney)
        {
            i_MinSum = i_Sum;
            i_MinMoney = i_TmpMoney;
        }
        i_Sum = 0;
        //V 3 Row
        for (int i_Col = 0; i_Col < ia_2DArr.GetLength(1); i_Col++)
        {
            i_Sum += ia_2DArr[i_Col, i_Row];
        }

        i_TmpMoney = ia_Money[(i_Sum - 6)];
        if (i_TmpMoney <= i_MinMoney)
        {
            i_MinSum = i_Sum;
            i_MinMoney = i_TmpMoney;
        }

        //V 2 incline
        i_Sum = ia_2DArr[0, 0] + ia_2DArr[1, 1] + ia_2DArr[2, 2];
        i_TmpMoney = ia_Money[(i_Sum - 6)];
        if (i_TmpMoney <= i_MinMoney)
```

```
        {
            i_MinSum = i_Sum;
            i_MinMoney = i_TmpMoney;
        }
        i_Sum = ia_2DArr[0, 2] + ia_2DArr[1, 1] + ia_2DArr[2, 0];
        i_TmpMoney = ia_Money[(i_Sum - 6)];
        if (i_TmpMoney <= i_MinMoney)
        {
            i_MinSum = i_Sum;
            i_MinMoney = i_TmpMoney;
        }
    }
    return i_MinSum;
}
}
```

個人認為完成作業須具備觀念

需先學會建立方法的概念，方法裡如果有帶參數值最後必定會有回傳值(return)，再來就是一些針對題目的要求，去思考程式的架構，如題目要求直線、橫線、斜線相加做比較，程式碼就分成三塊去實現