

# 第2次作業-作業-HW2

學號：109111101

姓名：邱韋翔

作業撰寫時間：150 (mins，包含程式撰寫時間)

最後撰寫文件日期：2022/10/01

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

先建立一個專門用來儲存炸彈的一維陣列，在建立一個二維陣列代表邊界大小，其中一維陣列指定 `int` 二維陣列指定 `char`，這樣方便後續用 `for` 迴圈時可以運用 `for` 迴圈的特性快速取出一維陣列裡的元素。建立完之後先用 `for` 迴圈讓二維陣列全部填滿 `0`，那再根據一維陣列存入炸彈的位置運用 `for` 迴圈填入進二維陣列裡。那再來就是去判斷，只要有炸彈(`*`)的周圍都要 `+1` 所以先寫出要判斷的二維陣列位置，再根據該位置先進行數字的轉換，因為原先二維陣列全部都是字元(`char`)，那會遇到炸彈周圍也有炸彈的情況，所以我加了一行判斷式如果該位置不是`*`則執行`+1`的動作，如果是則不做任何動作。最後在列印全部的二維陣列，下段程式碼為使用後結果：

```
public partial class Bomd : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        //填入0
        int[] ia_MIndex = new int[10] { 0, 7, 13, 28, 44, 62, 74, 75, 87, 90 };
        char[,] ia_Map = new char[10, 10];
        for (int i_Row = 0; i_Row < 10; i_Row++)
        {
            for (int i_Col = 0; i_Col < 10; i_Col++)
            {
                ia_Map[i_Row, i_Col] = '0';
            }
        }
        //填入*
        for (int i_Ct = 0; i_Ct < ia_MIndex.Length; i_Ct++)
        {
            int i_Row = ia_MIndex[i_Ct] / 10; //0
            int i_Col = ia_MIndex[i_Ct] % 10; //0
            ia_Map[i_Row, i_Col] = '*';
            //Response.Write(ia_Map[i_Row, i_Col]);
        }
    }
}
```

```
}  
//訪問*周邊  
for (int i_Ct = 0; i_Ct < ia_MIndex.Length; i_Ct++)  
{  
    int i_Row = ia_MIndex[i_Ct] / 10; //0  
    int i_Col = ia_MIndex[i_Ct] % 10; //0  
    //左上  
    if ((i_Row - 1) >= 0 && (i_Col - 1) >= 0)  
    {  
        if (ia_Map[i_Row - 1, i_Col - 1] != '*')  
        {  
            int i_Num = Convert.ToInt32(ia_Map[i_Row - 1, i_Col - 1])  
            i_Num++;  
            ia_Map[i_Row - 1, i_Col - 1] = Convert.ToChar(i_Num);  
        }  
    }  
    //上  
    if ((i_Row - 1) >= 0)  
    {  
        if (ia_Map[i_Row - 1, i_Col] != '*')  
        {  
            int i_Num = Convert.ToInt32(ia_Map[i_Row - 1, i_Col]);  
            i_Num++;  
            ia_Map[i_Row - 1, i_Col] = Convert.ToChar(i_Num);  
        }  
    }  
    //右上  
    if ((i_Row - 1) >= 0 && (i_Col + 1) < 10)  
    {  
        if (ia_Map[i_Row - 1, i_Col + 1] != '*')  
        {  
            int i_Num = Convert.ToInt32(ia_Map[i_Row - 1, i_Col + 1])  
            i_Num++;  
            ia_Map[i_Row - 1, i_Col + 1] = Convert.ToChar(i_Num);  
        }  
    }  
    //左  
    if ((i_Col - 1) >= 0)  
    {  
        if (ia_Map[i_Row, i_Col - 1] != '*')  
        {  
            int i_Num = Convert.ToInt32(ia_Map[i_Row, i_Col - 1]);  
            i_Num++;  
            ia_Map[i_Row, i_Col - 1] = Convert.ToChar(i_Num);  
        }  
    }  
}
```

```
        ia_Map[i_Row, i_Col - 1] = Convert.ToChar(i_Num);
    }
}

//右
if ((i_Col + 1) < 10)
{
    if (ia_Map[i_Row, i_Col + 1] != '*')
    {
        int i_Num = Convert.ToInt32(ia_Map[i_Row, i_Col + 1]);
        i_Num++;
        ia_Map[i_Row, i_Col + 1] = Convert.ToChar(i_Num);
    }
}

//左下
if ((i_Row + 1) < 10 && (i_Col - 1) >= 0)
{
    if (ia_Map[i_Row + 1, i_Col - 1] != '*')
    {
        int i_Num = Convert.ToInt32(ia_Map[i_Row + 1, i_Col - 1]);

        i_Num++;
        ia_Map[i_Row + 1, i_Col - 1] = Convert.ToChar(i_Num);
    }
}

//下
if ((i_Row + 1) < 10)
{
    if (ia_Map[i_Row + 1, i_Col] != '*')
    {
        int i_Num = Convert.ToInt32(ia_Map[i_Row + 1, i_Col]);
        i_Num++;
        ia_Map[i_Row + 1, i_Col] = Convert.ToChar(i_Num);
    }
}

//右下
if ((i_Row + 1) < 10 && (i_Col + 1) < 10)
{
    if (ia_Map[i_Row + 1, i_Col + 1] != '*')
    {
        int i_Num = Convert.ToInt32(ia_Map[i_Row + 1, i_Col + 1]);
        i_Num++;
        ia_Map[i_Row + 1, i_Col + 1] = Convert.ToChar(i_Num);
    }
}
```

```
        }  
    }  
}  
//最後訪問全部的陣列  
for (int i_Row = 0; i_Row < 10; i_Row++)  
{  
    for (int i_Col = 0; i_Col < 10; i_Col++)  
    {  
        Response.Write(ia_Map[i_Row, i_Col]);  
    }  
    Response.Write("<br />");  
}  
}  
}
```

---

## 個人認為完成作業須具備觀念

須先了解題目的要求，在了解 for 迴圈的語法和 if else 語法及邏輯運算子符號的運用，還有建立陣列的語法，以及型別的轉換(例如: char 、 int )，還有一些換行的技巧會運用到HTML語法