

程式設計第七次作業

Due: 2019/5/25 23:00

※注意事項：請依照課程網站內所公告之“作業檔案命名規則與規定”進行作業檔案命名以及繳交作業，未依照規定將斟酌扣分。

本次作業共有三題，第一題、第二題為問答題，第三題為程式撰寫。

專案僅須建立第三題的專案即可。

問答題以 WORD、記事本或是 PDF 檔案繳交皆可，檔案名稱以該題的題號為檔案名稱，如第二題的檔案名稱即為 HW07_02.docx。請壓縮在壓縮檔內，一同繳交。

●第一題：[\(30%\)](#)

參考下列程式碼中，依序回答下列問題：

(1) #2, #3 兩行程式碼的意義。

(2) 若 #3 沒有寫的話，compile 會發生錯誤，請問為什麼？以及有哪些解決的方法呢？

(3) #1, #2 有兩種 include 的方法（意指“xxx”及 <xxx> 兩種），請問這兩種方法有何差異。

```
#include "stdafx.h"           // #1
#include <iostream>             // #2
using namespace std;          // #3
```

```
int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Hello world!";
    return 0;
}
```

●第二題：[\(Bonus 10%\)](#)

MyPtrStruct 是一個存放指標的結構，m1, m2 為 MyPtrStruct 結構在主程式中的實體(instance)。在主程式中，我們將 m1 內的 ptr 指標實體化，並且賦予初值後，再令 m1 和 m2 相等。接著修改 m1 的值。

程式碼如下列所述，請編譯以下程式碼並回答問題：

1. 確認 m1, m2 在修改 m1 的值後，m2 輸出的值與 m1 是否會相同？並說明為什麼。
2. 程式碼中 delete [] m2.ptr; 會造成程式錯誤，為什麼？

```
struct MyPtrStruct
{
    int* ptr;
};

int _tmain(int argc, _TCHAR* argv[])
{
```

```

MyPtrStruct m1, m2;
    // 實體化 m1 內的指標，並賦予初值
m1.ptr = new int[5];
cout << "assign m1 data, and apply m2 = m1" << endl;
for(int i=0; i<5; i++)
    m1.ptr[i] = i*5;

m2 = m1; // 使得 m2 和 m1 相等

// 輸出 m1, m2 的值
cout << "m1 : " << endl;
for(int i=0; i<5; i++)
    cout << m1.ptr[i] << " ";
cout << endl;
cout << "m2 : " << endl;
for(int i=0; i<5; i++)
    cout << m2.ptr[i] << " ";
cout << endl;

// 重新修改 m1 的值
cout << "re-assign m1 data" << endl;
for(int i=0; i<5; i++)
    m1.ptr[i] = i*10;

    // 輸出 m1, m2 的值
cout << "m1 : " << endl;
for(int i=0; i<5; i++)
    cout << m1.ptr[i] << " ";
cout << endl;
cout << "m2 : " << endl;
for(int i=0; i<5; i++)
    cout << m2.ptr[i] << " ";
cout << endl;

delete [] m1.ptr; //清除 m1 的指標
delete [] m2.ptr; //清除 m2 的指標，會導致程式出錯

return 0;
}

```

●第三題：(70%)

根據作業五第三題改寫主程式，定義一個新的自訂資料型態，此自訂資料型態的成員包含指向二維動態陣列的指標、矩陣的行與列之值三個成員，宣告如下：

```
struct MyImage
{
    double **Data;
    unsigned int rows;
    unsigned int columns;
};
```

撰寫以下數個函數完成原題目的要求。函數包含：

1. 建立 MyImage 成員的函式，請重載此函數，
其一限制為單引數且無回傳值，在函數內呼叫“亂數產生函式”產生行列值，並建立二維動態陣列
另一函數限制為三引數且無回傳值，直接由主程式給予行列大小，在函數內建立二維動態陣列
2. 亂數產生矩陣行列值，引數輸入下界與上界，回傳隨機值，矩陣行列值介於 2 至 5 之間，使用者不需輸入上下界
3. 隨機建立矩陣元素值，此函數限制為單引數且無回傳值，在函數內給予二維陣列中各個元素 0 到 255 間的隨機整數值
4. 進行 MyImage 之折積的函數，此函數至多只能有三個引數，不限制回傳值形式，亦可不回傳
5. 求矩陣最大元素及其下標的函數
6. 列印出陣列元素的函數，此函數限制為單引數且無回傳值，引數傳入 MyImage，於函式中列印出 Matrix 的元素值
7. 歸還使用空間，此函數限制為單引數且無回傳值

只允許使用以上七種函數，不可再自行宣告其他函數，否則不予計分。

若函數有附加說明，未依照說明撰寫者，不予計分。

請以註解註明該函數屬於哪一個功能的函數。

※請勿使用標準樣板函式庫(Standard Template Library)或與題目無關之巨集指令※