

程式設計補充作業 S10-11

Due: 2021/5/24 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。

本次作業共有二題，第一題請繳交文字檔，第二題為程式撰寫。

●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：(70%)

克羅內克積(Kronecker product)在數學上為兩個任意大小矩陣間的運算，以 $\otimes$ 表示。克羅內克積是張量積(Tensor product)的一種特殊的形式。如果 A 為一個 $m \times n$ 的矩陣，而 B 為 $p \times q$ 的矩陣，則克羅內克積 $A \otimes B$ 為一個 $mp \times nq$ 的矩陣。

克羅內克積的定義：

$$A \otimes B = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & \cdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & \cdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}$$

運算範例：

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 1 \cdot 5 & 2 \cdot 0 & 2 \cdot 5 \\ 1 \cdot 6 & 1 \cdot 7 & 2 \cdot 6 & 2 \cdot 7 \\ 3 \cdot 0 & 3 \cdot 5 & 4 \cdot 0 & 4 \cdot 5 \\ 3 \cdot 6 & 3 \cdot 7 & 4 \cdot 6 & 4 \cdot 7 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$$

關於克羅內克積的詳細說明以及其他運算範例請自行上網至參考資料閱讀。

定義二個新的自訂資料型態，其一之自訂資料型態為儲存矩陣之資料型態 MyMatrix，其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；另一之自訂資料型態為記錄數值及其位置之資料型態 MyValue，其成員包含欲紀錄之數值、記錄其所在矩陣之位置(地址)的指標、其在矩陣內之位置(下標)。

```
struct MyMatrix
{
    double **Matrix;
    unsigned int rows;
    unsigned int columns;
};

struct MyValue
{
    double Data;
    MyMatrix* matrix;
    unsigned int row;
    unsigned int column;
};
```

利用動態配置兩個二維 double 矩陣(其中 A 矩陣為 $m \times n$ 矩陣、B 矩陣為 $p \times q$ 矩陣， m, n, p, q 可能不相等且介於 3 到 6 間，請分別利用亂數產生)，將兩個 double 陣列及矩陣的 rows 及 columns 以 MyMatrix 結構進行封裝，矩陣內的元素亦利用亂數產生(在-21.3 到+9.2 之間)。函數包含：

1. 建立 MyMatrix 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(MyMatrix)行列值(m, n, p, q)，引數輸入下界與上界，回傳隨機值，矩陣行列值介於 3 至 6 之間
3. 隨機建立矩陣(MyMatrix)內元素值，此函數限制為單引數且無回傳值，在函數內給予 MyMatrix 內二維陣列中各個元素-21.3 到+9.2 間的隨機值
4. 進行兩矩陣(MyMatrix)之克羅內克積($A \otimes B$)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(MyMatrix)中最小元素及其下標的函數(若矩陣最小元素有多筆，只須回傳一筆)，結果以 MyValue 進行封裝並 return by value 回傳 MyValue。
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 MyMatrix，於函式中列印出 MyMatrix 的元素值
7. 列印 MyValue 之函式，應輸出元素數值、其下標及其所在之矩陣(MyMatrix)之地址
8. 歸還 MyMatrix 使用空間，此函數限制為單引數且無回傳值

主程式呼叫以上函數計算並於視窗輸出以下內容：

1. 兩個原始矩陣，並標示出 A 矩陣與 B 矩陣；
2. 兩矩陣進行克羅內克積($A \otimes B$)後的矩陣；

3. 求矩陣最小元素及其下標，下標由 1 開始標記，Ex:最小值在[2][2]的位置，則應顯示為 (3,3)。

輸出均在主程式及列印出陣列元素函數中進行，換言之其他函數中不可出現 `cout` 或 `printf` 等。

不可以修改函數引數或是回傳值的形式，否則不予計分。

不可自行宣告其他函數，否則不予計分。

請為引數適當加入 `const`，保護資料不被更動。

※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※