

## 程式設計補充作業 S8

Due: 2021/5/1 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。

本次作業共有四題，第二題、第三題及第四題為問答題，第一題為程式撰寫。

專案僅須建立第一題的專案即可。

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 HWS8\_02.docx。請壓縮在壓縮檔內，一同繳交。

●第一題：[\(60%\)](#)

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。

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)，兩矩陣內的元素亦利用亂數產生(在 -21.3 到 +9.2 之間)。撰寫以下幾個函數

- 亂數產生矩陣行列值(引數輸入下界與上界，回傳隨機值)  
inline int random\_int(const int lower, const int upper);
- 亂數產生矩陣元素(引數輸入下界與上界，回傳隨機值)  
inline float random\_float(const float lower, const float upper);
- 建立二維動態陣列  
double\*\* create(const int rows, const int columns);

4. 隨機建立矩陣元素值，限制不可有回傳值，引數的格式請自行設計  
void create\_elements(.....);
5. 兩矩陣進行克羅內克積( $A \otimes B$ )，由 result 回傳結果  
void kronecker\_product(double \*\*result, const double \*const\*const matrix1, const int rows1, const int columns1, const double \*const\*const matrix2, const int rows2, const int columns2);
6. 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)。最小元素 return by value，下標的兩個元素以 pass by pointer 回傳，其餘引數格式請自行設計  
..... minimum( ..... );
7. 歸還陣列使用空間  
void release(double \*\*&matrix, const int rows);
8. 輸出函數，函數內列印矩陣的結果，引數格式請自行設計  
void output\_matrix( ..... );

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)。

輸出均在主程式及函數 **output\_matrix** 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※

●第二題：[\(10%\)](#)

請以文字檔案詳細描述第一題中函數 4 create\_elements、5 kronecker\_product、6 minimum、7 release，四個函數的引數及回傳值分別是以那種形式(value, pointer, reference)傳入或回傳及使用該形式的理由或是用處為何？

## ● 第三題：(15%)

試回答下列三個輸出的結果，並說明為什麼有這樣的結果。

```

void PassByValue(double a)
{
    a *= 10.0;
}

void PassByRef(double& a)
{
    a *= 10.0;
}

void PassByPointer(double* a)
{
    *a *= 10.0;
}

int _tmain(int argc, _TCHAR* argv[])
{
    double number = 10.0;
    PassByValue(number);
    cout << "PassByValue: " << number << endl;

    number = 10.0;
    PassByRef(number);
    cout << "PassByRef: " << number << endl;

    number = 10.0;
    PassByPointer(&number);
    cout << "PassByPointer: " << number << endl;

    return 0;
}

```

## ● 第四題：(15%)

請依序說明下列兩個 CreateArray 函式的回傳結果有何不同，並說明為什麼會有這樣的差別。

```

double* CreateArray1(double preset)
{
    double arr[10];

    for (int i = 0; i < 10; i++)
    {
        arr[i] = preset;
    }

    return arr;
}

double* CreateArray2(double preset)
{
    double* arr = new double[10];

    for (int i = 0; i < 10; i++)
    {
        arr[i] = preset;
    }

    return arr;
}

```