

在 `Transpose()` 函數中，因為轉置在數學上的定義是把矩陣中每個元素的行列交換，而稀疏矩陣的特點就是非零元素佔了很少部分，因此比起講義中基於行列的模式，我認為基於元素進行轉置操作的效率較高，將每個元素的行列值互換後再依照位置重新排序，

`FastTranspose()` 函數則是完全參照講義。

串流輸入運算子部分為避免元素以不符合預期的方式輸入造成運算錯誤，因此在輸入完畢後重新排序，維持元素排列的一致性。

加法運算子是依照位置，由 a_{11} 、 a_{12} …… a_{21} 、 a_{22} …… a_{n1} …… a_{nm} 的順序，依序相加各個位置相對應的元素。

乘法的部分，根據乘法矩陣原理，第 n 行與第 m 列各個元素相乘後的總合為相乘結果的 nm 位置的元素，因此以轉置過後的乘數矩陣進行操作，以結果矩陣的行列為準，依序尋找相對應的所需矩陣元素，相乘後加總即得結果矩陣的元素值。

為了完成位置的排序與比較，自行撰寫了 `pos_comp()` 函數，比較原則是行為先列為後，如有兩元素 a_{13} 與 a_{22} ，則定義前者位置較小，若另有兩元素 a_{31} 與 a_{32} ，定義前者位置較小。

加分題的內容為將程式修改為不以固定長度陣列儲存元素，以應對不同數量的輸入元素，因此改成使用不定常向量 `vector` 達成需求，並修改輸出串流運算子以取得正確所需參數。