Program Enviroment:

OS: win8.1

Compiler: GCC

Program Description:

<typedef struct>

建立一個稀疏矩陣表示法的元素結構，每個結構元素的內容有列數，行數及數值內容

<main>

首先以scanf函式抓取指定建立矩陣的行數和列數（若是行數和列數不是大於零，則印出錯誤提示後結束程式），並以此建立矩陣a和b（b為a的轉置，故其行數和列數與a相反）

並以巢狀迴圈重複讀取每列的數字，且以countValue指定當前讀到第幾個元素

（countValue在陣列開始時初始為1，因為稀疏矩陣的第一個元素用來儲存矩陣的行數、列數、非零元素個數）

若是讀取到的數字不為零，則將數字存入指定元素的value值中，並指定當前迴圈計算到的行列數給col值和row值中，並將countValue值加一

讀取的數字若為零則不存入稀疏矩陣，否則有違稀疏矩陣表示法的本意

當讀取完所有的輸入時，此時的countValue值應當是矩陣內所有非零元素的總個數，於是我將其指定給a和b矩陣第一個位置的value值

此時，因為已經處理好a矩陣內的所有內容，故呼叫fastTranspose函式而得到a矩陣轉置後的b矩陣

為了印出b矩陣的所有元素，以巢狀迴圈印出每列每行的元素，並又將countValue初始化為1，以計算目前指定到第幾個元素

為了印出稀疏矩陣表示法沒有儲存的元素零，採用以下策略：

讀取元素時，若是每列開始讀取到的元素（也就是每列第一個元素）row值和應該印出的行數不相等，則印出零

若是每行開始讀取到的元素col值和應該印出的列數不相等，則印出零

（因為列行數不相等代表表示法中沒有該值，但的確有該值）

除此之外，印出所有讀取到的元素，並每次將countValue加一

<fastTranspose>

仿照課本範例，但改以一個陣列rowPos來儲存轉置後的起始位置

（因為rowPos是為了指定轉置時，原本陣列的各元素需移動到的位置，rowPos陣列的長度為矩陣的元素個數）

首先判斷numTerms是否為小於或等於零的情況，防止矩陣非零時進行轉置運算

並對rowPos陣列的每個元素以「0」值初始化

再讀取原本矩陣的每個元素，計算某行中有幾個元素，並將結果指定給rowPos

至此，rowPos中的每個元素分別對應到原本矩陣中每行的元素數目

為了將rowPos轉為各元素需要位移的起始位置，以tempNext變數儲存當前處理的rowPos元素，並對此rowPos元素進行改寫，改寫規則如下：

若處理的是第一個rowPos元素（rowPos[0]），則改寫為1，因為第一個元素轉置後依然是第一個位置

若處理的不是第一個rowPos元素（rowPos[0]），則以temp變數儲存上一個位置儲存的tempNext（也就是上一個rowPos未進行改寫前的值），並以改寫後的上一個位置rowPos元素（rowPos[i-1]）加上temp

至此，每個rowPos儲存的資料將是由1開始，累加轉置前每行對應的元素個數的數目，也就是轉置後該行開始的第一個元素必須移動到的位置

最後以rowPos進行矩陣轉置，以變數j讀取每個元素對應到的rowPos，進行轉置，由於rowPos的位置資訊只是針對該行第一個元素，所以再處理完之後，將該對應的rowPos元素加一，以便再讀取到該行第二個元素時，可以移動其到正確的位置