

作業二 第二題

一、題意：

題目想要同學回答的是最後一行：**Obtain an algorithm to determine the value of $a[i][j]$, $0 \leq i, j < n$ from the array b .** “Obtain an algorithm” 就是指 “寫出一段程式碼”，而不用寫出完整的可執行程式碼。”to determine the value of $a[i][j]$ ” 是要找出 $a[i][j]$ 的值，也就是給定任意 i, j 的值，只要是在允許的範圍內 ($0 \leq i, j < n$)，找出該 element 的值，這只一個數字，而不是要找整個 a 矩陣，也不是要找 b 。最後，”from the array b ” 是說剛剛的 $a[i][j]$ 值，要從 array b 裡面找出來，也就是說，array b 已經存在了，同學不需要再去做 initialization 的動作，畢竟 initialization 是整個程式的另一個部分，而不是這個 ”找 element” 的 algorithm 所需要做的。

如果同學們有仔細觀察課本的程式碼，你們會發現課本幾乎沒有給過 initialization（除非那個 function 就是要做 initialization 的動作），這是因為每個 function 都有它自己的功能，如果它不是用來做 initialization 的，那就不應該碰 initialization 的部分。但是課本會宣告該 function 所需要的 “資料結構”，也就是在使用 function 前的變數宣告。這也就是助教們希望同學們回答的格式。

二、參考解答：

程式碼主要分為以下幾個部分：

- 變數宣告
- 函式部分
- 主要計算部分（如何由 i, j 轉換為正確的 index）
- 錯誤檢查（error-checking）

```
#define max_element 1000
double b[max_element];
int n; // matrix dimension = n x n
double item; // used to store a[i][j]

// return 1 for success, return 0 for failure
int findTridMatrixItem(int n, int i, int j, double *item)
{
    int pos = 2*i + j; // convert a[i][j] to b[pos]

    // indices exceed matrix dimension
    if ((i < 0) || (j < 0) || (i >= n) || (j >= n))
        return 0;

    if ((j < i-1) || (j > i+1))
        *item = 0;
    else
        *item = b[pos];
    return 1;
}
```

三、評分標準：

由於助教們事先沒能和同學解釋好這題的題意和解答的需求，這題會改的比較寬鬆，只要大致符合題意就可以給分，助教們也在此向同學們說一聲抱歉！基本上，不論是做從 b, i, j 找 a ，還是從全部的 a 找全部的 b ，或是全部的 b 找全部的 a ，只要程式正確都可以給分。錯誤檢查的部分，由於事先沒有說，所以有做的同學會加分，沒做的不會扣分。

- 滿分 = 50 (55) 分
- 有做錯誤檢查 = +1 ~ 5 分 bonus (沒做不扣分)
- 主要計算部分 ($pos = 2*i + j$ 或任何類似的方法) = 20 分
- 變數宣告 = 15 分
- 傳回想要的值 (或矩陣) = 5 分
- 其他 (如註解，程式碼正確... 等等) = 10 分
- 其他的小錯誤，每個視情形扣 1 ~ 2 分

四、常見問題：

- 測試是否相等要用雙等號 `==`，而一個等號是給值 (assignment) 用的。
- 很多同學說明不夠完整，又沒有程式碼來說明，助教實在不知道你給的答案到底想要解釋什麼。
- 很多同學寫的 `pseudo-code` 不像是 `pseudo-code`，更需要寫的更明確一點，更容易實現一點。例如，你不可以直接寫“把那個答案算出來”這種 `pseudo-code` (尤其是這題就是要考你們如何算，而你的 `pseudo-code` 又沒講)；大家可以看看課本的 `program4.22` (page 169)，你們會發現，其實 `pseudo-code` 還是和原來的 C 程式碼很像，而並不是說寫 `pseudo-code` 就可以隨便寫 (尤其是 `pseudo-code` 的 `for` 和 `while` 用法不可以改得太離譜！)
- 公式在推導過程中要記得做簡化，而不是把一堆式子都加減乘除在一起，變成長長的一行之後就沒事了。
- 程式碼一定要記得寫註解，不要以為自己現在看得懂就算了，幾個月後或幾年後你不一定還記得，而要別人看你沒註解的程式碼更是一件殘忍的事 (助教已經看你們的程式看到快發瘋了@@)
- 所有用過的變數一定要記得宣告。雖然你們只是在寫一小段程式碼，而不是一個完整的可執行程式碼，但如果你們不宣告，天知道你的變數 k 是 `int` 還是 `char*` 還是什麼奇怪的東西。
- 很多同學在使用 `for` 迴圈時常常會少算一個 (有時是多算一個)，例如：當陣列 a 大小是 n ，則要看到每一個 a 的元素要用 `for (i=0; i < n; i++)` 而不是 `for (i=0; i <= n; i++)`，也不是 `for (i=1; i < n; i++)`，(假設 i 是 `int`)。而矩陣是 $n \times n$ 大小時，最右下角的元素是 $(n-1, n-1)$ 而不是 (n, n) 。
- 很多人喜歡用變數名稱小寫 L ，雖然這是程式語言所允許的，但這是很不好的寫法，因為小寫 L 和數字 1 很像 (至少在大部分的字型都是如此)，這會讓讀你的程式的人很難去分辨你到底在寫什麼，其他的例子包括：“`llji`”，“`oO0`”，“`mnrn`”(例：“`arn`”和“`am`”，第一個是小寫 `ARN`，第二個是小寫 `AM`)

五、評分助教：

這題是由陳亮宇所做的評分，對於分數有問題的同學可以在我的 office hour 來找我，我的 office hour 是週二下午 17：00～19：00 在資電館 326 教室，我對每一份作業評分都有存檔。請盡量“不要”用 email 來問有關於作業評分的事。