

資結 練習二 報告

(一) 簡介:

任務一:

(1)讀檔：

先用 Inputfile 這個 function 讀檔，如果讀檔成功就用雙迴圈把檔案內容存到一個二維陣列裡面。

二維陣列的列數記載為 rowMax、行數記載為 columnMax 以此來設定地圖的範圍。

(2)走迷宮

宣告兩個陣列 visitedX、visitedY 分別存取走過的點之座標，四個 function 來判斷上下左右是否可走，如果是遇到 O(障礙)、V(走過的路)、或是超過陣列範圍則代表不可走。

宣告布林 Check：是否全部走完的部分

宣告布林 Cannot find：讓後面的 print 能區別有沒有找到 G

從陣列的第一個元素(0,0)當起點，如果不為 G(目標)，就進入迴圈，迴圈裡呼叫上下左右四個 function 來找尋 G 的位置，走過的位置會設成 V，遇到無路可走的地方用 visitedX、visitedY 進行倒退，找到 G 或是 check = true(走完所有可走的路)時就跳離迴圈。

Print :

1.成功找到 G：第一個矩陣印出改變過後的二維陣列，第二個矩陣把 visitedX、visitedY 存取的路徑變成 R 後印出。

2.沒有找到 G：只印出改變過後的二維陣列

任務二:

與任務一大致相同，任務二新增一個變數 i 來紀錄找到幾個 G，當 i 跟使用者輸入的 G 個數一樣時或走完地圖所有可走的路，就跳出迴圈。

當還沒找到足夠的 G 時，就重複做任務一，當找齊 G 或走完地圖上能走的路就印出結果。

Print :

1.成功找齊 G：第一個矩陣印出改變過後的陣列，第二個矩陣把前面找到使用者輸入個數的 G(目標)的路線通過 visitedX 和 visitedY 紀錄的 printdata 路線改成 R 後印出。

2.沒有找齊 G 或沒找到任何一個 G：只印出改變過後的陣列。

防呆：如果使用者輸入非數字那就再請他輸入一次，如果輸入的數字不在

1~100 之間則告訴他超過範圍並請他再輸入一次，直到輸入 1~100 間的數字。

任務三:

新增一個布林 check 來測試是否比儲存長度短

新增一個 length，找到第一個 G 後紀錄走了幾步並存於 length，之後繼續找 G

時若大於 length 則不繼續走，若找到第二個 G 步數少於 length 則取代它。

Print：

1. 如果有成功找到目標的最短路線:第一個矩陣印出改變後的矩陣，第二個矩陣印出離起點最近的目標的路線，最後印出 length。
2. 如果沒有成功找到目標的最短路線(就是代表沒有辦法走到任何目標): 就會印出改變後的矩陣和找不到去目標的路線的句子。

防呆：與任務二相同

心得：

這次的三個任務都不簡單，一開始聽到題目完全沒有想法，在網路上找了好多相關資料後腦中才擁有這些概念，整理好資料開始要寫的時候我們遇到第一個難題--讀檔，讀檔的部分就讓我們需要花一點時間，上網研究了一下也了解了 fstream 的使用方法，中間還遇到一個問題就是 Openfile 是放 string 進去 open 會出現 compile error 後來是打 demo.c_str()(整個字串加上.c_str()) 將 String 物件轉換成 C 語言形式的字串常數，才解決了這個問題。

在做任務一的時候，我們邊界沒設定好所以找了很久的 Bug，找到後才知道要用陣列的行列最大值去當邊界才是安全的方法。

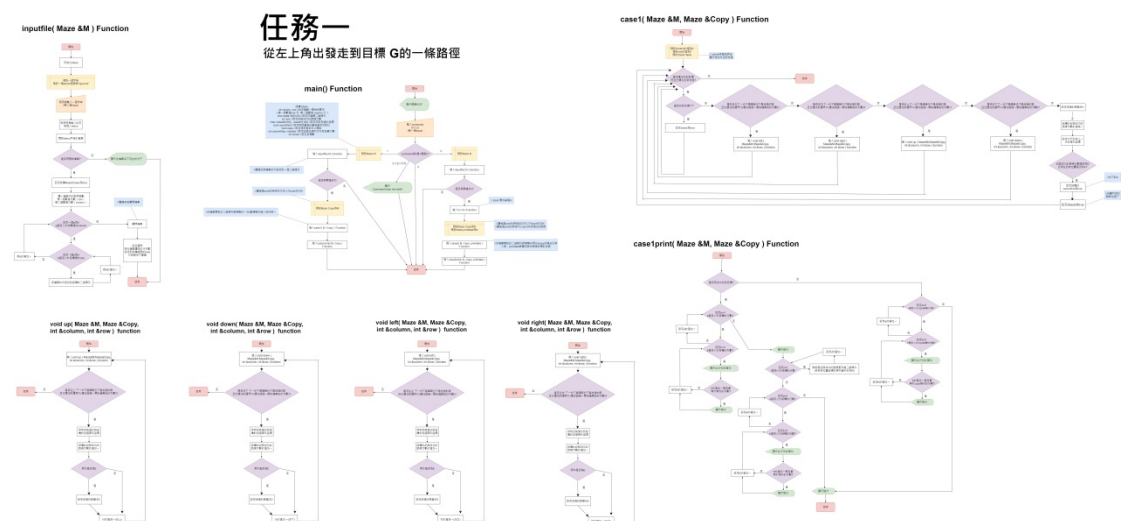
任務二則是在找到 G 後迴圈無法繼續走，幸好最後有想到將 G 先設為另一個變數，方能完成。但沒過多久就發現問題，當測試老師的 Demo 跟 show 檔才知

道老師的走法是一個方向遇到牆才會轉向，原來我們寫錯了！

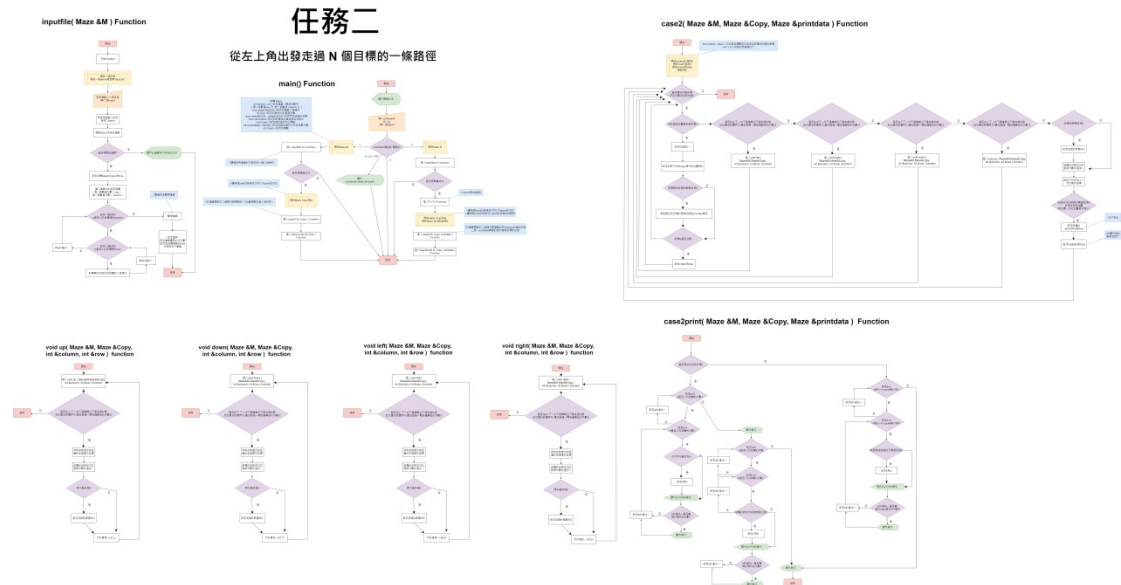
花了一些時間，終於想到加上遞迴，用遞迴新寫四個方向的 function 讓它能透過遞迴完成迷宮老師正確的走法，不過任務三怎麼寫都有些 input 跟老師的最短路線不一樣，想了很久都理不出頭緒，最後才知道要用 BFS 廣度優先搜尋法，但時間已經截止了也來不及改程式碼，因此任務三比較遺憾無法每個 input 都找到最短路線。

(二) 圖示：

任務一：



任務二：



(三) 解說

影片連結: https://www.youtube.com/watch?v=NFMiT_M70sI