1. 簡介

在main()中輸入可能的指令command，如果command為數字 ‘1’，讀進迷宮文件檔名稱，處理任務一，找出走到目標G的路徑；如果command為數字 ‘2’，讀進文件檔名稱，再讀入目標G的數量，處理任務二，找出走過N個目標G的路徑；如果是 ‘0’，則程式執行結束；如果都不是，請使用者再輸入一次指令。

1. 任務一

利用class Maze{}宣告一迷宮amaze，amaze是由struct maz{}所組成的二維陣列，根據先前讀進的檔案名稱讀取檔案內容讀進迷宮amaze中，路徑從左上角(0,0)開始找起，呼叫amaze.Go()分別依循方向右,下,左,上確定是否找到終點，過程中適時更改布林值r(route),v(visited)，表示經過的證明，最後以char型別的state表示該格子的狀態，最後分別以 ’V’及 ’R’印出所有走過的位置和成功走到目標G的路徑(若沒有成功路徑，輸出曾走過的位置即可)。

心得

讀檔的部分確實有點複雜，網路上也有許多範例，後來選擇老師推薦的作法，其中開檔的作法滿特別的，

string filename ;

FILE \*infile = NULL ;

filename = "input" + command + ".txt" ;

infile = fopen( filename.c\_str(), "r" ) ;

剩下的就是讀進二維陣列中了。

1. 任務二

與任務一相同，初始化陣列(v=false, r=false)後讀入迷宮內容，一樣從(0,0)開始，以迴圈找終點，重複設定x,y為上一個終點，直到成功找齊N個目標(若不存在任何路徑，則輸出第一個矩陣 ‘V’)。

1. 流程圖
2. 問答

動態陣列是指在編譯時不能確定陣列長度，程式在執行時需要動態分配記憶體空間的陣列，因此可以有效的節省使用記憶體空間；另外在存取資料時老師使用的是更有效率的字串存取，我們的方法則是字元存取。