Program Enviroment:

OS: win8.1

Compiler: GCC

Program Description:

<int board[8][8]>

宣告一個8\*8的整數矩陣，用來儲存騎士巡邏棋盤的移動位置

<int ifExit (int i,int j)>  
檢查輸入的i,j值（位置）是否在board的範圍內，且該位置指定的board內容是否為零（為零則表示騎士尚未走過此位置）

若是，則回傳非零值(1)；若否，則回零值傳

<main>

首先按照課本上指定的順序設定ktmove1和ktmove2兩陣列，並對board陣列進行初始化（初始為0）

並宣告nexti、nextj兩個陣列來儲存每一步可以移動的方向位置和exits陣列儲存該方向位置的下一步可以走的方向個數

接著進行輸入提示及讀取起始點(i,j)

呼叫ifEXit判斷起始點(i,j)是否在board的範圍內且該位置內容為0零，並為了保險，將該位置設為零（因為初始位置尚未移動，並不算移動軌跡）。

若是ifExit判斷初始點位置錯誤，將會輸入提示進行錯誤警告。

並進行一64次的迴圈（因騎士最多走到64步）。在迴圈中，每次都將nexti、nextj、exits陣列及npos、min、l初始化為0

npos用來紀錄該點可以走的方向個數

min則紀錄起始點可以走的方向內，出口數最小的位置

l紀錄每個可以走的方向位置。

並以一個循環八次的陣列來計算當前位置可以走的方向（以變數k指定），每次呼叫ifExit判斷k變數指定的方向是否在合法範圍內。

如果是則以l變數依序指定nexti和nextj儲存該位置，

並將npos變數加一，儲存可以移動的方向數

再來只要對npos進行判斷，即可簡化狀況

若npos為零，代表該位置沒有任何可以走的位置，跳出當前迴圈，並印出失敗提示

若npos為一，代表只有一個方向可以走，該方向位置儲存在nexti和nextj陣列的第一個元素（nexti[0]、nextj[0]）中，故指定min為0

若npos為其他情況（1<npos<=8），則將變數temp賦值為8（temp是用來儲存最小的exits陣列元素，故先指定元素中最大的可能，才可進行比較）、變數min賦值為0（min是用來儲存最小的exits陣列元素的索引值，也就是l，故先指定為第一個元素索引值）

並進行一迴圈，每次檢查某一方向的下一步出口數，此迴圈只進行npos次（因為只有npos個方向）

為了檢查下一步出口數，迴圈每次進行一八次迴圈，呼叫ifExit檢測出口數，並以exits紀錄相對應位置的出口數

每次一得到新的exits元素，則比較temp與它。若是temp較大，代表當前位置的出口數較小，將temp指定為當前位置的出口數，以進行下一次比較；把min指定為當前位置對應到的索引值

再來選取min指定的nexti、nextj元素，就是出口數最小的可移動方向。若是該方向元素不為0，則將i、j修改為該位置，並紀錄該位置為第幾次移動的步數

由於for迴圈的特性，最後一次的m值將會是總共步數加一，故先將m減一

判斷減一後的m是否為64步，如果是則印出巡邏成功

最後印出騎士巡邏的軌跡，和總共步數