上機習題 #3 (stack之應用)

題目：西洋棋騎士(knight)走完棋盤的順序

說明：

一個n\*n的西洋棋盤上，騎士(knight)置於某位置(x,y)，撰寫一個程式以便找出此騎士(n2-1)次移動的順序，而每個格子只拜訪過一次。(使用stack，而不使用recursion)。西洋棋騎士可能移動之方位共有8個如下：

j

j

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 |  | K8 |  | K1 |  |
| 2 | K7 |  |  |  | K2 |
| 3  i  i  j |  |  | K |  |  |
| 4 | K6 |  |  |  | K3 |
| 5 |  | K5 |  | K4 |  |

上圖中，K為目前騎士所在之位置，K1至K8為可能移動之位置。假設K所在之位置為(i,j)，則移動後之位置差值如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  | i之差值 | j之差值 | 位置 |
| K1 | -2 | 1 | (i-2,j+1) |
| K2 | -1 | 2 | (i-1,j+2) |
| K3 | 1 | 2 | (i+1,j+2) |
| K4 | 2 | 1 | (i+2,j+1) |
| K5 | 2 | -1 | (i+2,j-1) |
| K6 | 1 | -2 | (i+1,j-2) |
| K7 | -1 | -2 | (i-1,j-2) |
| K8 | -2 | -1 | (i-2,j-1) |

可以利用嘗試錯誤的方法(backtracking)解決此問題。騎士在某位置時，可以選擇某條路往前走，走到不能繼續前進時（所有可能移動的位置，均已嘗試過），則退回上一個位置，再選擇其他路徑。由於騎士往前走，而將來又可能退回，故可利用stack記錄騎士所走過的路徑。

Stack的每一個元素應至少記錄三項資料如下：

i 軸之值、j軸之值、該位置已經走過的方向編號

例如：n=3

initial:

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |

從(1,1)出發，3步之後：

|  |  |  |
| --- | --- | --- |
| 1 | 0 | 0 |
| 0 | 0 | 2 |
| 3 | 0 | 0 |

上圖的2與3分別表示第二次與第三次拜訪的位置，0表示尚未拜訪過。此時stack的內容如下：

|  |
| --- |
|  |
| 3,1,x |
| 2,3,6 |
| 1,1,3 |

Stack的內容，2,3,6表示i=2，j=3，方向為6（即K6）。

撰寫此程式，除了需要有一個stack外，還需要一個n\*n的陣列(array)。剛開始，陣列所有元素均設定為0，此時的0表示該位置尚未被拜訪過。如果騎士走了三步，則依序在其拜訪過的位置記錄為1、2、3。將來若發現此條路徑不通需要退回時，則需將這些位置重設為0，表示日後騎士尚可由其他路徑拜訪該位置。

輸出：

以(0,0)為起點，分別印出n=1、2、3、4、5、6之情形。

如果無解，印出"no solution"。如果有解，則在各個格子內填入拜訪的順序 m，代表該位置是於第m步被拜訪的。例如n=5，可印出：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 20 | 17 | 12 | 3 |
| 16 | 11 | 2 | 7 | 18 |
| 21 | 24 | 19 | 4 | 13 |
| 10 | 15 | 6 | 23 | 8 |
| 25 | 22 | 9 | 14 | 5 |

對於有解的n，印出任一組解即可，不必印出所有的解。

**注意：本習題不得使用recursive方法撰寫程式，必須使用stack。**