

## Assignment #3 (An application of stack)

Goal : Test if a knight can travel around a chess board

Guide :

On an  $n*n$  chess board, a knight is initially placed on some position  $(x,y)$ . You are asked to write a program to report the following  $(n^2-1)$  moves of this knight, so that each position on the chess board is visited exactly once. You should complete this assignment with a stack, not recursion. The movement of knight on a chess board is as follows:

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

Suppose the knight stands on K, then K1 to K8 are eight possible moves of this knight. Assume K is on position  $(i,j)$ , then the locations of K1 to K8 can be written as follows.

	diff to i	diff to j	Location
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	$(i-2,j-1)$

You can solve this problem by **trial and error**. When the knight is standing on some position K, you can test all possible moves K1 to K8 one by one. If all possible moves fail, then return the previous location from which the knight moved to K, and keep testing other possible moves. Since the knight keep going forward and backward, you can use a stack to record the path of the knight. Each element in the stack should keep three

different values at least: they are i, j, and the direction that has been tested.  
Taking n=3 for example

initial:

0	0	0
0	0	0
0	0	0

3 steps that begin at (1,1)

1	0	0
0	0	2
3	0	0

The label '2' and '3' represent the locations that the knight visit by its second and third move, respectively, and the label '0' means the location has not been visited yet. The stack for the above example is as follows:

3,1,x
2,3,6
1,1,3

The numbers 2,3, and 6 represent i=2, j=3, and the direction K6, respectively.

To write this program, you also need an n\*n array, whose elements are initially set to zero. For a knight that has moved 3 steps, each visited locations are labeled with 1, 2, 3, which record the order of visit. When a knight returns to its previous location, these positions should be reset to 0, so that the knight can still visit these locations from other paths.

Output :

Set (0,0) as the starting point of the knight. You have to print the path for n=1, 2, 3, 4, 5, and 6. If there is no solution, then print "no solution". If there is a solution, then print each label on the chess board.

Taking n=3 for example, the labels are

1	4	7
6		2
3	8	5

However, for n=3 there should be no solution.

**Note: You must complete this assignment with stack.**

**A recursive program will not be accepted.**