Practice S07P07: Flood!

http://www.comp.nus.edu.sg/~cs1010/4_misc/practice.html

Week of release: Week 8

Objectives: 2D array, function with pointer parameters

Created by: Lim Wei Zhong

Task statement:

A flood is coming to robot land! The land is a rectangle of unit squares, but it has uneven elevation. Each unit square has a unique elevation, and initially contains exactly one robot. The robots are all given a command to go to high ground. However, they were only programmed to follow the following procedure: among the adjacent squares (sharing borders), if there is a square with a higher elevation, pick the one with the highest elevation, and go there. Otherwise, stay on the current square. They repeat this until they do not move. We assume that each square has enough space for all the robots.

19	18	16	4	3
14	1	12	9	5
6	10	11	15	0
8	7	2	13	17

(a)

For example, given the elevations in table (a),

- The path of the robot at (1,2) is (1,2) → (0,2) → (0,1) → (0,0)
- The path of the robot at (3,1) is (3,1) → (2,1) → (2,2) → (2,3)

The owner of these robots wants to save all of them! He wants you to write a program to tell him how many robots he needs to rescue given that the robots have already stopped moving between squares and after that the flood will come and reach a certain elevation f. He needs to rescue a robot if and only if it ended up on a square with an elevation lower than f.

For example, given the elevations in table (a), the final number of robots in each square is as shown in table (b).

8				
			7	
1				4

(b)

- The robots at (0,0) are from (0,0), (0,1), (0,2), (0,3), (1,0), (1,1), (1,2), (2,0).
- The robots at (2,3) are from (0,4), (1,3), (1,4), (2,1), (2,2), (2,3), (3,1).
- The robots at (3,0) are from (3,0).
- The robots at (3,4) are from (2,4), (3,2), (3,3), (3,4).

If the flood reaches an elevation of 17, the robots in (3,0), whose elevation is 8, and (2,3) whose elevation is 15, needs to be rescued. Therefore, the answer is 1+7 = 8.

Write a program **flood.c** to compute the number robots that needs to be rescued. You should include a function `findEndPos()` which computes the final position of a particular robot given its initial position. For example, given the elevations in table (a) and the initial position (1,2), this function should return (0,0).

Challenge: **Solve this problem using recursion!**

(You do not need to include the required function for this challenge.)

Sample runs:

```
Enter dimensions of the land: 3 3
Enter elevations:
1 2 3
4 5 6
7 8 9
Enter flood level: 9
Number of robots to be rescued: 0


Enter dimensions of the land: 3 3
Enter elevations:
4 7 1
3 6 8
9 2 5
Enter flood level: 9
Number of robots to be rescued: 6

Enter dimensions of the land: 3 3
Enter elevations:
7 4 8
5 0 3
6 2 9
Enter flood level: 8
Number of robots to be rescued: 4

Enter dimensions of the land: 4 5
Enter elevations:
19 18 16 4 3
14 1 12 9 5
6 10 11 15 0
8 7 2 13 17
Enter flood level: 17
Number of robots to be rescued: 8

Enter dimensions of the land: 5 5
Enter elevations:
13 23 5 20 6
14 10 11 22 12
4 15 7 16 9
9 3 17 8 18
0 21 2 1 24
Enter flood level: 17
Number of robots to be rescued: 4
```