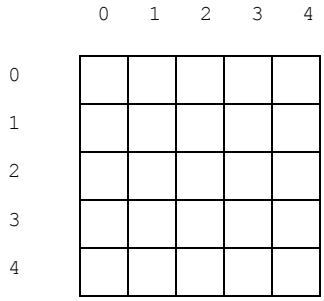
University of Massachusetts Dartmouth
Department of Electrical and Computer Engineering

ECE 161 — Submit name: chess.cpp
Project 3 - chess — Due: See http://ece161.viall.org

A knight is a chess piece which moves in a unique way. A knight must always move in an "L" shape; that is two squares in one direction (horizontal or vertical) and one square in the other direction. For this program, use 5 x 5 chess board.



Write a program which uses recursion to determine a sequence of moves whereby a knight begins at the upper left corner of a 5 x 5 chess board, and in exactly 24 moves, visits each square exactly once.

Your output may be in one of two forms; the first six moves are shown for each:

[0,0], [1,2], [2,4],[4,3], [3,1], [1,0], [2,2], etc

OR

```
0   x   x   x   x
5   x   1   x   x
x   x   6   x   2
x   4   x   x   x
x   x   x   3   x
```

`x` is number between 7 and 24

Note, this is only a sample. Depending on how you order your moves, you may get a different sequence.

Useful routines:

set(r,c);	marks square r, c as visited
reset(r,c);	unmarks square r, c as visited
visited(r,c);	returns true if square r, c visited
boarddone();	returns true if all squares visited
move(r, c);	attempt to find next move from square r, c this is the recursive routine. This routine may or may not return a value, depending on your algorithm.

Other hints:

For debugging, each time a move is made, print a message like:

```
Moving to square rr cc
```

Each time a move is undone, print a message like:

```
Backing out move to rr cc
```

Note that printing each move slows down the program considerably (1 second vs. 4.5 minutes). Use this only to determine that moves are being made correctly.

If the knight moves off the board for any given move, you can treat it as though that square is already occupied.