Junior Knights Program: Minesweeper

For this assignment, you will implement a version of the popular game, Minesweeper. Instead of a 8x8 board, implement a 5x5 board. When the game begins, display a gameboard where all squares are covered:

```
  0 1 2 3 4
0 _ _ _ _ _
1 _ _ _ _ _
2 _ _ _ _ _
3 _ _ _ _ _
4 _ _ _ _ _
```

For our purposes, randomly fill 5 of the 25 cells on the board with bombs. For each turn, ask the user to enter in the row and column of their move (0-4 for both). If that cell is occupied with a bomb, the user has lost the game. At this point you can print out a version of the board, completely uncovered. Otherwise, uncover the contents of that square. The contents of the square will be a single integer signifying the total number of adjacent squares with bombs. For example, if the user enters row 1, column 4, we could show the following board to the user:

```
  0 1 2 3 4
0 _ _ _ _ _
1 _ _ _ _ 2
2 _ _ _ _ _
3 _ _ _ _ _
4 _ _ _ _ _
```

If the user is lucky enough to pick a square that has zero adjacent bombs, automatically recursively clear all adjacent un-cleared squares. As an example, if the user's second move is to uncover cell row 4, column 0, the result could be:

```
  0 1 2 3 4
0 _ _ _ _ _
1 _ _ _ _ 2
2 _ _ _ _ _
3 1 1 1 2 _
4 0 0 0 1 _
```

The game either ends when the user uncovers a bomb, or when the user uncovers all squares EXCEPT for the ones that contain the bombs. If the latter occurs, the user wins!!! If you'd like, you can time the user and show their winning time at the end of the game. (Of course, in a text-based system like this, that time will be very slow =))

The full uncovered game board for the running example above is:

```
  0 1 2 3 4
0 * * 2 1 1
1 3 3 3 * 2
2 1 * 2 2 *
3 1 1 1 2 2
4 0 0 0 1 *
```

Implement error checking as you see necessary.