CSCI-101 Introduction to Programming

Lab 9 - Tuesday

Log onto cs.bridgewater.edu. Change your working directory to the **labs** directory in your repository. Create a subdirector (in labs) named **lab9**. Change your working directory to **lab9**. When complete with this lab, please push your code to GitHub and update your **README.md** file to provide the path to the **lab9** directory and your Java source code.

Write a program in a file named **Wombat.java** that implements the following game.

When run, the program creates a 10x10 2D matrix of characters. The elements in the matrix are initialized to equal sign (=) characters. This matrix is the playing field for the game.

The program places strings of capital Xs, representing pits, either vertically, horizontally, or diagonally in the array. Add one string of Xs of length 6, one of length 5, one of length 4, one of length 3, and one of length 2.

If you were to print the contents of the 2D array after it has been initialized with pits it should look something like this (with your pits possibly at different locations).

```
X X X X X X = = = =
= = = = = = = = = =
= X = = = X X = = =
= = X = = = = = = =
= = = X = = = = = =
= = = = X = = = = =
= = = = = X = = X =
= = = = = = = = X =
= = X X X X = = X =
= = = = = = = = = =
```

The game is played as follows. The player starts in the cell at row 10, column 1 and has to maneuver, using the arrow keys, up to the cell at row 1, column 10 without hitting a pit. If the player hits a pit they have to start over.

When the game begins the program will repeatedly do the following:

1. Show the player a game field that shows where he has travelled (using + characters) and the pits that he has fallen into, but not the pits that he has not fallen into.
2. Tell the player how many pits surround his current location.
3. Ask the player to make a move by pressing the characters w (up), a (left), s (down), and (d) right.
4. If the player moves into an invalid location, tell the player it is an invalid move and go to step 1.
5. Move the player.
6. If the player moves into a pit, restart the player at row 10, column 1 and go to step 1.
7. If the player moves to the final destination cell, congratulate the player and end game.

One goal of this assignment is to make the main() method as short as possible. Below are a list of method signatures that one could possibly use to define methods that do certain tasks. Feel free to use these or create your own.

- static void initGameField(char[][] m)
- static void printGameField(char[][] m)
- static int numPits(char[][] m, int row, int col);
- static boolean isValidMove(int curRow, int curCol, int nextRow, int nextCol)