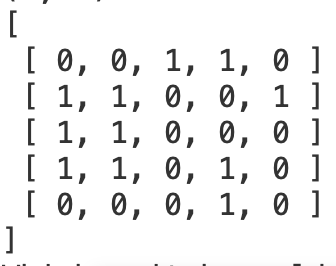
**LightsOut!**

Within the provided starter file, write the console-based game "LightsOut!"

1. First, **play the game** [**here**](https://www.logicgamesonline.com/lightsout/). The goal is to "turn off" all the lights (aka make the grid go dark) in the smallest number of clicks possible.
2. Now, let’s take a look at the starter file. We’ve provided the following functions:
   1. makeStartingConfiguration(): To make debugging and testing the game easier, we’ve hardcoded the board below for you to start with:

****

This 5x5 board is composed of 1’s and 0’s. The **1’s represent a square where the light is “on” and the 0’s represent a square where the light is “out”** (so the player’s goal is to end with a 5x5 board composed only of 0’s).

* 1. printSolution(): Also to make debugging and testing easier, we’ve provided the solution to the board above. The solution is a list of (row, col) pairs which correspond to the squares on the board; when those squares are selected in order, all the lights should be out (aka the board should be composed of 0’s).
  2. print2dList(): From the course notes, this prints out a 2D list in a semi-cute way.
  3. play(): **This is the function that you will be modifying**. Please do not modify any of the other given functions described above; feel free to write your own helper functions and call them inside play()**.** When this function is called, the board we provided prints out already.

1. Let’s modify play()!
   1. After the board we’ve provided prints out, you want to **ask for the user’s** [**input**](https://www.kosbie.net/cmu/spring-21/15-112/notes/notes-getting-started.html#ConsoleInput) to find out which square they want to click. We recommend asking the user to format their input as **row col**.
   2. Now that you have the row and col that the user wants to click, you’ll want to **flip the lights (0 → 1, 1 → 0) in the correct squares** (the one square that was selected and the four squares directly north, south, east, west to it)
   3. Then, you want to repeat steps 3a and 3b again and again and again (what could you use here so the steps repeat continuously?) until all the lights are out and the user has won (how can you check the board for a win?)!
2. Now you have a playable game of LightsOut! Add the following features to make the user experience even better.
   1. Handle invalid user input (i.e. non-integer inputs, inputs that are formatted incorrectly, etc.).
   2. Print out the board every time the user clicks a square so it’s clear which lights have been flipped.
   3. Print out nice messages while asking for input and after the user has won.
3. And there you have it! You’ve created a functioning game; add or create some new boards and you can play it yourself. Feel free to add some more additional features if you have time.
4. **Remember to submit your completed code to Autolab under the hw5-lab assignment**.