COW – Puzzler

For this COW, you will be manipulating a 2D array of Puzzle Pieces. The PuzzlePiece class methods that you will need to use are as follows:

Each of the following methods returns the key on the puzzle piece. When the puzzle is correctly formed, each adjacent set of sides share a unique key. Edges that are on the outside of the entire puzzle, and thus don't share a side with any other piece, have a key of -1. The anchor piece is located in the upper left hand corner at index 0, 0. The anchor piece's outside edges have a key of -2.

```
public int getEast(){}
public int getNorth(){}
public int getWest(){}
public int getSouth(){}
```

This method rotates the puzzle piece one time clockwise:

```
public void rotateClockwise(){}
```

In each level complete the listed methods in the Puzzler Class:

Level 1

Name: `rotateAll`
Input: `PuzzlePiece [][] thePuzzle`
Output: `nothing`
Action: rotate each of the puzzle pieces clockwise one time

Level 2

Name: `randomlyRotate`
Input: `PuzzlePiece [][] thePuzzle`
Output: `nothing`
Action: rotate each of the puzzle pieces clockwise a random number of times so that each one is facing a random direction, potentially including its original direction

---

Level 3

Name: `fixRotation`
Input: `PuzzlePiece [][] thePuzzle`
Output: `nothing`
Action: rotate each piece in the puzzle until they are all facing the correct direction. You may assume that the pieces are in the correct location.

Level 4

Name: `randomizeOrder`
Input: `PuzzlePiece [][] thePuzzle`
Output: `nothing`
Action: randomize the location of each puzzle piece

Level 5

Name: `fixPlacement`
Input: `PuzzlePiece [][] thePuzzle`
Output: `nothing`
Action: puts each of the puzzle pieces in its correct location. You may assume they are all facing the correct direction.

Level 6

Name: `randomizeCompletely`
Input: `PuzzlePiece [][] thePuzzle`
Output: `nothing`
Action: randomize the location and direction of each puzzle piece

Name: `solve`
Input: `PuzzlePiece [][] thePuzzle`
Output: `nothing`
Action: moves and rotates the pieces until they are in the correct order and orientation. You may not assume that any of the pieces are in the correct location or facing the correct direction.