Exam Set 22E Solutions
AP Computer Science A

Name ____ Period__________

1. Refer to the following code:

```
   int [][] zorro = new int[3] [4];

     for(int row=0; row<zorro.length; row++)
     {
          for(int col=0; col<zorro[row].length; col++)
          {
               zorro[row][col] = row * 2;
          }
     }
```

(a) Indicate the values of the `zorro` array above. The numbers in the table indicate the indices of the array.

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

/2

2. The `Grid` class below prints a two-dimensional array of doubles. The `Grid` constructor should accept a two dimensional array of any size as a parameter. The `showGrid` method should print out the contents.

(a) Declare a two-dimensional array called `doubleArray`, but do not initialize it.
(b) Write the constructor that accepts a two-dimensional array as a parameter then assigns the array to `doubleArray`.
(c) Write the `showGrid` method which prints out the contents of the two dimensional array

```
public class Grid{

     public double[][] doubleArray;

     public Grid(double[][] da){
          doubleArray = da;
     }

     public void showGrid(){

          for(int row = 0; row < doubleArray.length; row++{
               for(int col = 0; col < doubleArray[row].length; col++){
                    //prints the current row
                    System.out.print(doubleArray[row][col]);
               }
          System.out.println();//goes to the next line
          }
     }
}
```

/4

__________/18

3. In the driver class below,

(a) Declare and initialize a 3x3 two-dimensional array called `doubleArray`
(b) Create a `Grid` object called `doubleGrid`.
(c) Print the contents of `doubleGrid.`

```
public class gridDriver{

  public static void main(String[] args){


     double[][] doubleArray = new double[3][3];
     Grid doubleGrid = new Grid(doubleArray);
     doubleGrid.showGrid();



  }

}
```

/3

Score __________/18

4. The `LightBoard` class models a two-dimensional display of lights, where each light is either on or off, as represented by a Boolean value. You will implement a constructor to initialize the display and a method to evaluate a light.

```
public class LightBoard
{
     /** The lights on the board, where true represents on and false
      *  represents off.
      */

     private boolean[][] lights;

     /** Constructs a LightBoard object having numRows rows and numCols columns
      *  Precondition: numRows > 0, numCols > 0
      *  Postcondition: each light has a 40% probability of being set to on
      */

     public LightBoard(int numRows, int numCols)
     {     /* To be implemented in part (a)  */    }

     /** Evaluates a light in row index row and column index col
      *  and returns a status as described in part (b).
      *  Precondition: row and col are valid indexes in lights.
      */

     public boolean evaluateLight(int row, int col)
     { /* to be implemented in part (b) */ }

      // There may be additional instance variables, constructors, and methods not
shown.
}
```

(a) Write the constructor for the `LightBoard` class, which initializes lights so that each light is set to on with a 40% probability. The notation `lights[r][c]` represents the array element at row `r` and column `c`.

Complete the `LightBoard` constructor below.

```
/** Constructs a LightBoard object having numRows rows and numCols columns.
 * Precondition: numRows > 0, numCols > 0
 * Postcondition: each light has a 40% probability of being set to on.
 */

public LightBoard(int numRows, int numCols) {

     lights = new boolean[numRows][numCols];
     for(int rows = 0; rows < lights.length; rows++{
          for(int cols = 0; cols < lights[rows].length; cols++){
               double r = Math.random();
               lights[row][col] = (r <= .40);
          }
     }
}
```

/4

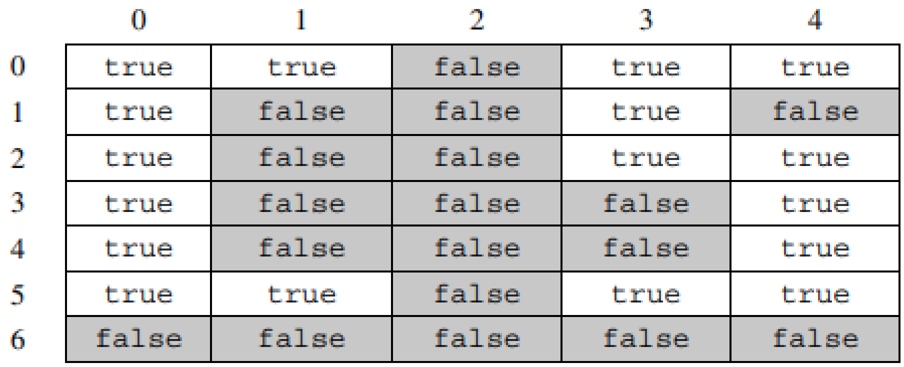
Score __________/18

(b) Write the method `evaluateLight`, which computes and returns the status of a light at a given row and column based on the following rules.

1. If the light is on, return false if the number of lights in its column that are on is even, including the current light.
2. If the light is off, return true if the number of lights in its column that are on is divisible by three.
3. Otherwise, return the light's current status.

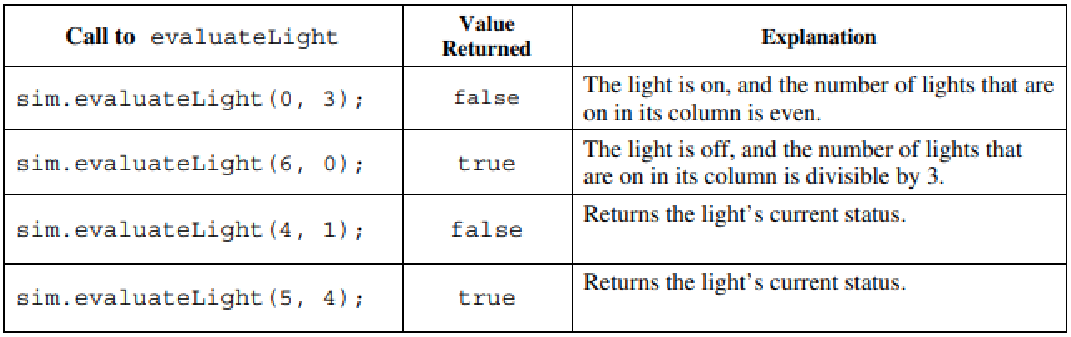
For example, suppose that `LightBoard sim = new LightBoard(7, 5)` creates a light board with the initial state shown below, where true represents a light that is on and false represents a light that is off. Lights that are off are shaded.

lights



	0	1	2	3	4
0	true	true	false	true	true
1	true	false	false	true	false
2	true	false	false	true	true
3	true	false	false	false	true
4	true	false	false	false	true
5	true	true	false	true	true
6	false	false	false	false	false

Sample calls to `evaluateLight` are shown below.



Call to `evaluateLight`	Value Returned	Explanation
`sim.evaluateLight(0, 3);`	`false`	The light is on, and the number of lights that are on in its column is even.
`sim.evaluateLight(6, 0);`	`true`	The light is off, and the number of lights that are on in its column is divisible by 3.
`sim.evaluateLight(4, 1);`	`false`	Returns the light's current status.
`sim.evaluateLight(5, 4);`	`true`	Returns the light's current status.

Score __________/18

Class information for this question

```
public class LightBoard
private boolean[][] lights
public LightBoard(int numRows, int numCols)
public boolean evaluateLight(int row, int col)
```

Complete the evaluateLight method below.

```
 /** Evaluates a light in row index row and column index col and returns a status
  * as described in part (b).
  * Precondition: row and col are valid indexes in lights.
  */

public boolean evaluateLight(int row, int col)

public boolean evaluateLight(int row, int col) {

     int numOn = 0;

     for (int r = 0; r < lights.length; r++) {

          if (lights[r][col]) {
               numOn++;
          }
     }

     if (lights[row][col] && numOn % 2 == 0) {
          return false;
     }

     if (!lights[row][col] && numOn % 3 == 0) {
          return true;
     }

     return lights[row][col]; }

}
```

/5

Score ___________/18