

1. Write an enhanced for loop which prints the contents of this 2D double array in columns and rows with a space between each value as shown to the right.

```
double[][] nums = //contains valid data;
```

```
for (double[] row: nums)
    for (double n: row)
        System.out.print(n + " ");
    System.out.println();
}
```

(Sample output only; do not assume **nums** has these dimensions.)

```
1.1 2.3 1.5 6.2 0.9 9.1
1.1 2.3 1.5 6.2 0.9 9.1
1.1 2.3 1.5 6.2 0.9 9.1
```

2. Create a 2D int array that is 5 rows by 10 columns:

```
int [][ ] nums = new int [5][10];
```

3. Write the default value of each type of 2D array:

```
double: 0.0
int: 0
String: null
Frog: null
```

4. Write an array initializer that creates a 2D String array that contains these values arranged as shown (each letter is its own element):

```
c a t
d o g
```

```
String [][ ] letters = { { "c", "a", "t" }, { "d", "o", "g" } };
```

5. Without looking at your repl from Monday, given the Light class to the right, write code to create a 5 by 100 2D Light array named **partyLights** of instantiated Light objects.

```
Light [][ ] partyLights = new Light [5][100];
for (int r = 0; r < partyLights.length; r++)
    for (int c = 0; c < partyLights[0].length; c++)
        partyLights[r][c] = new Light();
}
```

```
public class Light
{
    private String color;

    public Light()
    {
        int c = (int) (Math.random() * 3);
        if (c == 0)
            color = "red";
        else if (c == 1)
            color = "green";
        else
            color = "blue";
    }

    public String getColor()
    {
        return color;
    }
}
```

6. Write a static int method **columnCount** that searches a 2D Light object array of any size and returns how many columns in the array happen to be all the same color.

```
public static int columnCount (Light [][] check)
{
    int count = 0;
    for (int c = 0; c < check[0].length; c++)
    {
        boolean allSame = true;
        for (int r = 1; r < check.length; r++)
        {
            if (!check[r][c].getColor().equals(check[r-1][c].getColor()))
                allSame = false;
        }
        if (allSame) count++;
    }
    return count;
}
```

7. Write a static boolean method "eachColorPresent" which accepts a 2D array of Light objects and returns true if the 2D array contains at least one of each color (red, green blue), false otherwise.

```
public static boolean eachColorPresent (Light [][] check)
{
    boolean foundRed = false;
    boolean foundGreen = false;
    boolean foundBlue = false;
    for (int r = 0; r < check.length; r++)
    {
        for (int c = 0; c < check[r].length; c++)
        {
            String color = check[r][c].getColor();
            if (color.equals("red")) foundRed = true;
            if (color.equals("green")) foundGreen = true;
            if (color.equals("blue")) foundBlue = true;
        }
    }
    return foundRed && foundGreen && foundBlue;
}
```