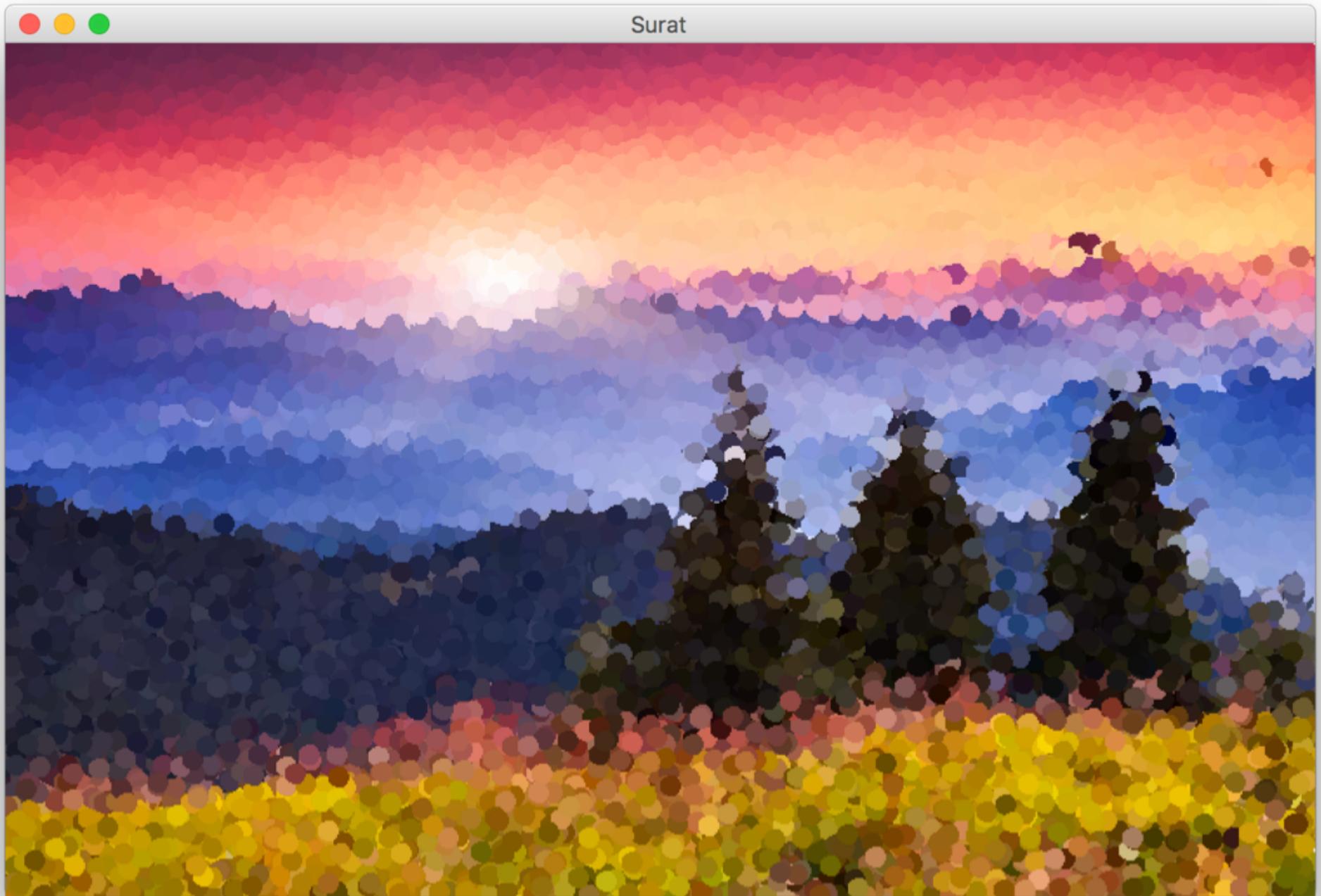




The Matrix

Chris Piech

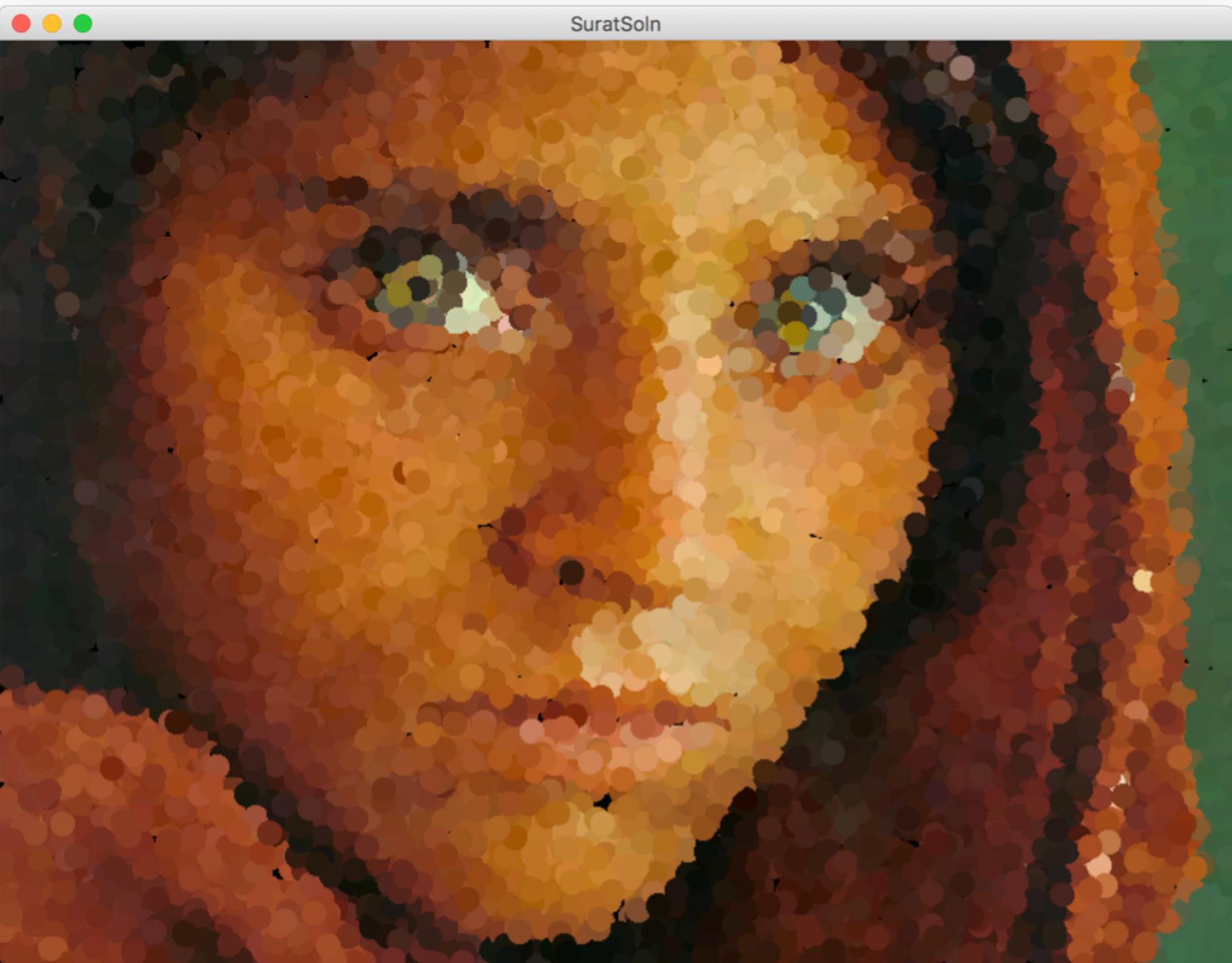
CS106A, Stanford University



Surat

Piech, CS106A, Stanford University





SuratSoln

Piech, CS106A, Stanford University



What does this say?

53‡†305)) 6* ; 4826) 4‡•) 4‡) ; 806* ; 48†8¶
60)) 85 ; 1‡ (; : ‡*8†83(88) 5*† ; 46 (; 88*96*
? ; 8) *‡ (; 485) ; 5*†2 : *‡ (; 4956*2 (5*‐4) 8¶
8* ; 4069285) ;) 6†8) 4‡‡ ; 1 (‡9 ; 48081 ; 8 : 8‡
1 ; 48†85 ; 4) 485†528806*81 (‡9 ; 48 ; (88 ; 4 (
‡?34 ; 48) 4‡ ; 161 ; : 188 ; ‡? ;

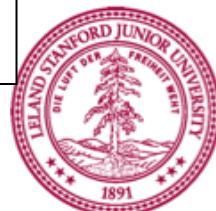


Poe's Cryptographic Puzzle

53‡†305)) 6* ; 4826) 4‡•) 4‡) ; 806* ; 48†8¶
60)) 85 ; 1‡ (; : ‡*8†83 (88) 5*† ; 46 (; 88*96*
? ; 8) *‡(; 485) ; 5*†2 : *‡(; 4956*2 (5*-4) 8¶
8* ; 4069285) ;) 6†8) 4‡‡ ; 1 (‡9 ; 48081 ; 8 : 8‡
1 ; 48†85 ; 4) 485†528806*81 (‡9 ; 48 ; (88 ; 4 (
‡?34 ; 48) 4‡ ; 161 ; : 188 ; ‡? ;

8	33
;	26
4	19
‡	16
)	16
*	13
5	12
6	11
(10
†	8
1	8
0	6
9	5
2	5
:	4
3	4
?	3
¶	2
-	1
•	1

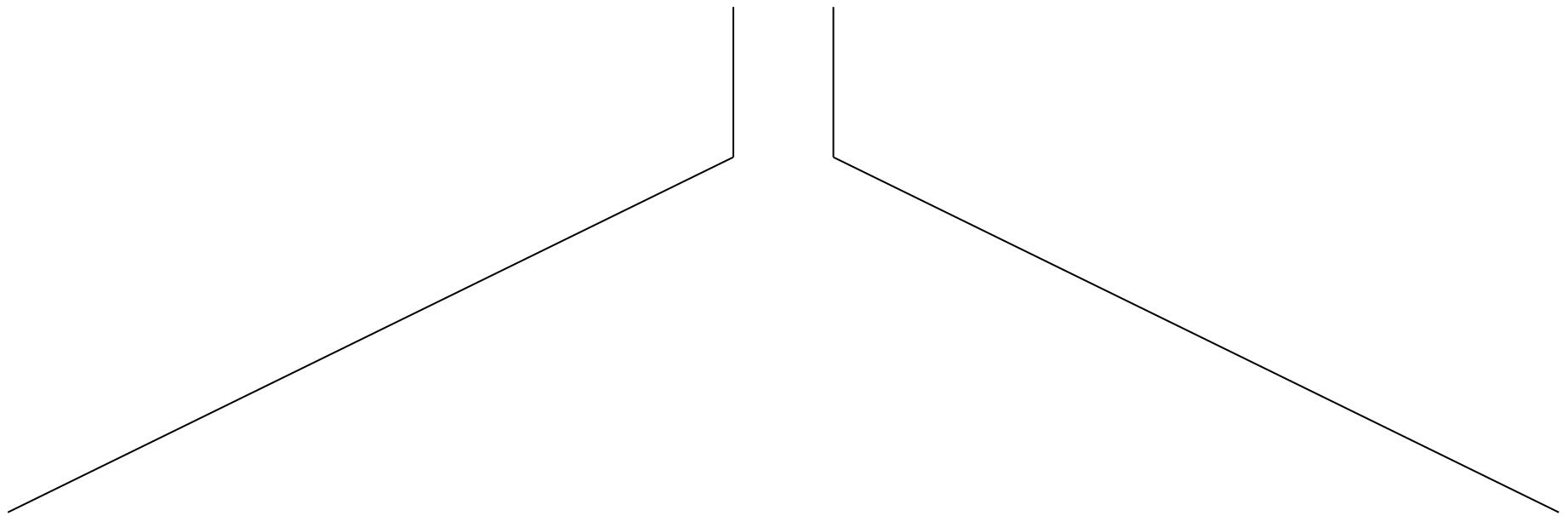
A GOOD GLASS IN THE B SHOPS HOSTED BY THE DEV
BOSSEAT FORTY ONE DEGREES AND THIRTEEN MIN
UTES NORTH EAST STAND BY NORTH MAIN BRANCH SEVEN
ENTH STREET BEASTS DESIRED FROM THE LEFT EYE OF
THE BEAST SHEADABEEN FROM THE TREE STAR
OUGHT THE SHOT GIFTY FEET OUT



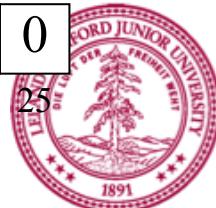
Implementation Strategy

The basic idea behind the program to count letter frequencies is to use an array with 26 elements to keep track of how many times each letter appears. As the program reads the text, it increments the array element that corresponds to each letter.

TWAS BRILLIG



1	1	0	0	0	0	1	0	2	0	0	2	0	0	0	0	1	1	1	0	0	1	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25



What does code for this look like?

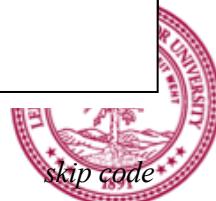
Count Letter Frequencies

```
import acm.program.*;  
  
/**  
 * This program creates a table of the letter frequencies in a  
 * paragraph of input text terminated by a blank line.  
 */  
public class CountLetterFrequencies extends ConsoleProgram {  
  
    public void run() {  
        println("This program counts letter frequencies.");  
        println("Enter a blank line to indicate the end of the text.");  
        int[] frequencyTable = new int[26];  
        String line = readLine();  
        countLetterFrequencies(frequencyTable, line);  
        printFrequencyTable(frequencyTable);  
    }  
  
    /* Displays the frequency table */  
    private void printFrequencyTable(int[] table) {  
        for (char ch = 'A'; ch <= 'Z'; ch++) {  
            int index = ch - 'A';  
            println(ch + ": " + table[index]);  
        }  
    }  
}
```



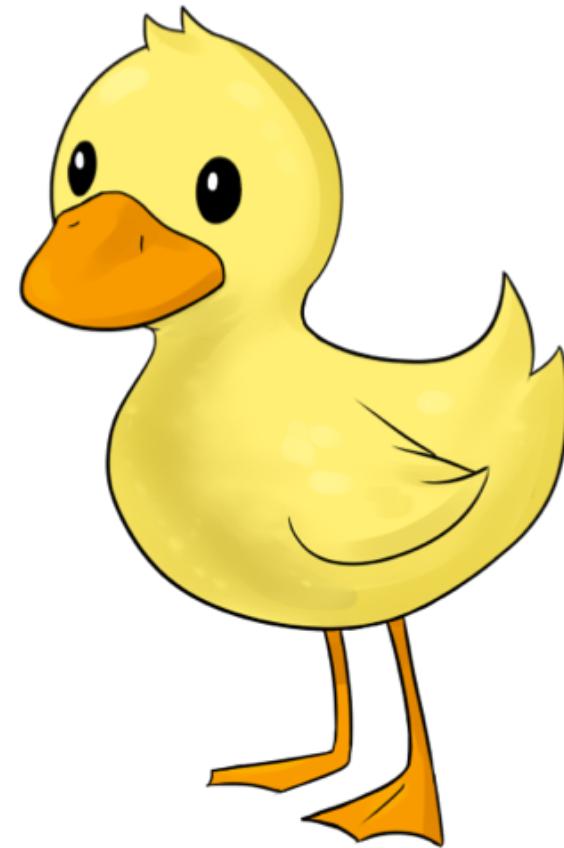
Count Letter Frequencies

```
/* Counts the letter frequencies in a line of text */
private void countLetterFrequencies(int[] table, String line) {
    for (int i = 0; i < line.length(); i++) {
        char ch = line.charAt(i);
        if (Character.isLetter(ch)) {
            int index = Character.toUpperCase(ch) - 'A';
            table[index]++;
        }
    }
}
```



Dim the lights

Value:
Yellow

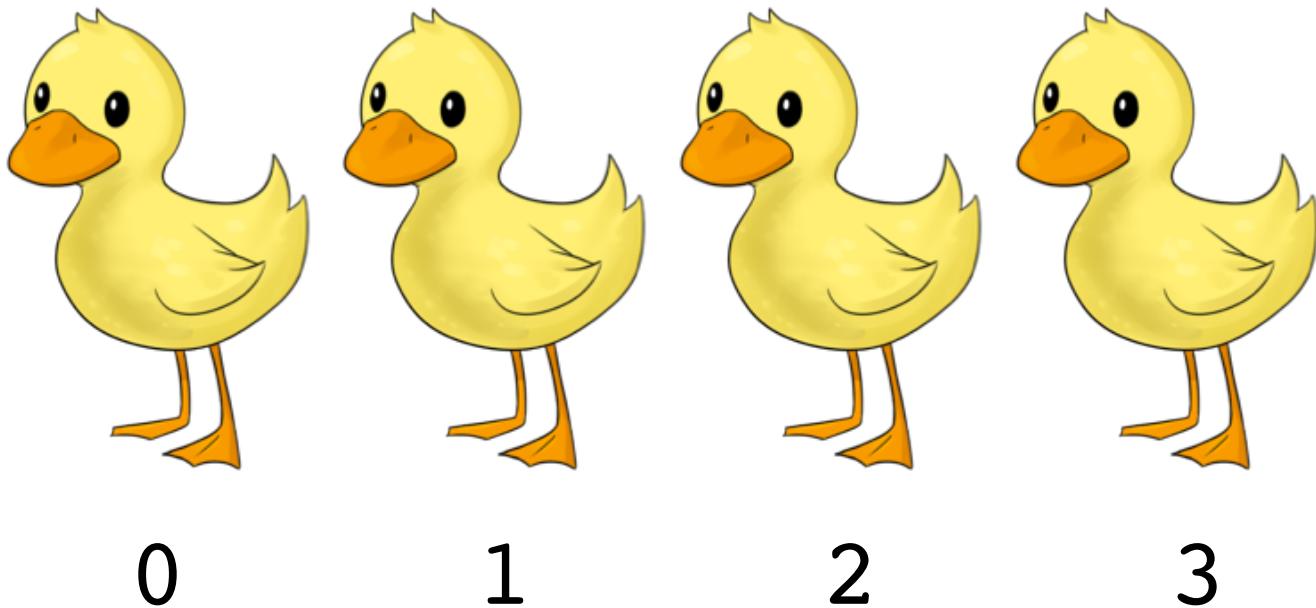


Type:
Duck

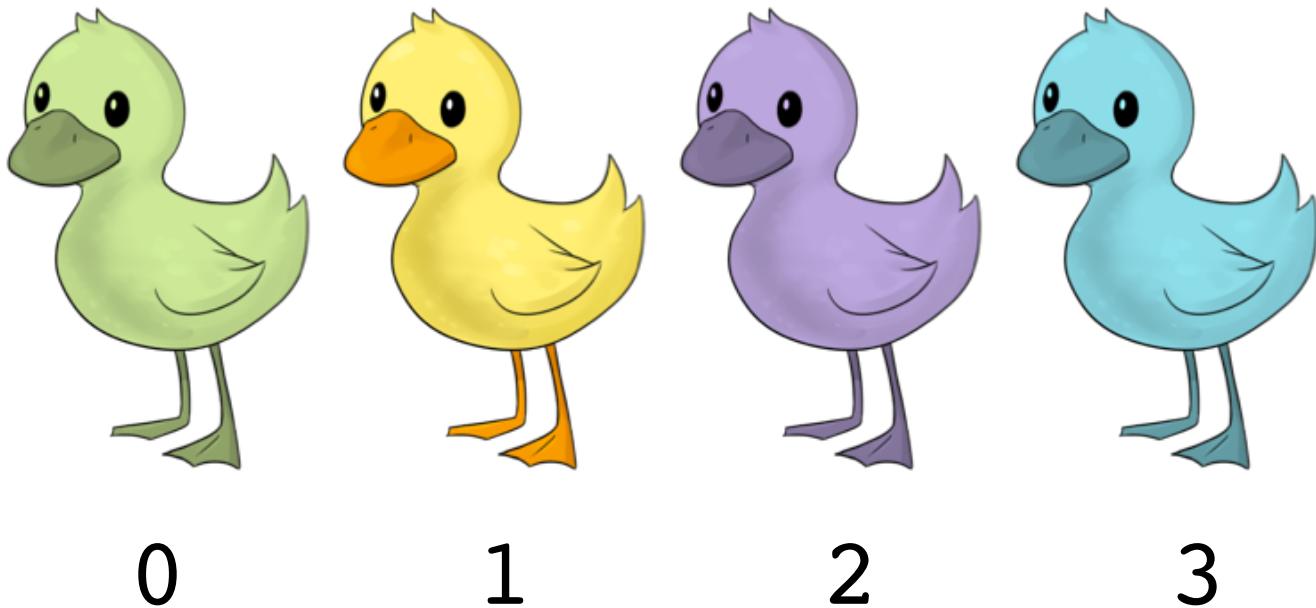
Metaphor for
a bucket in memory

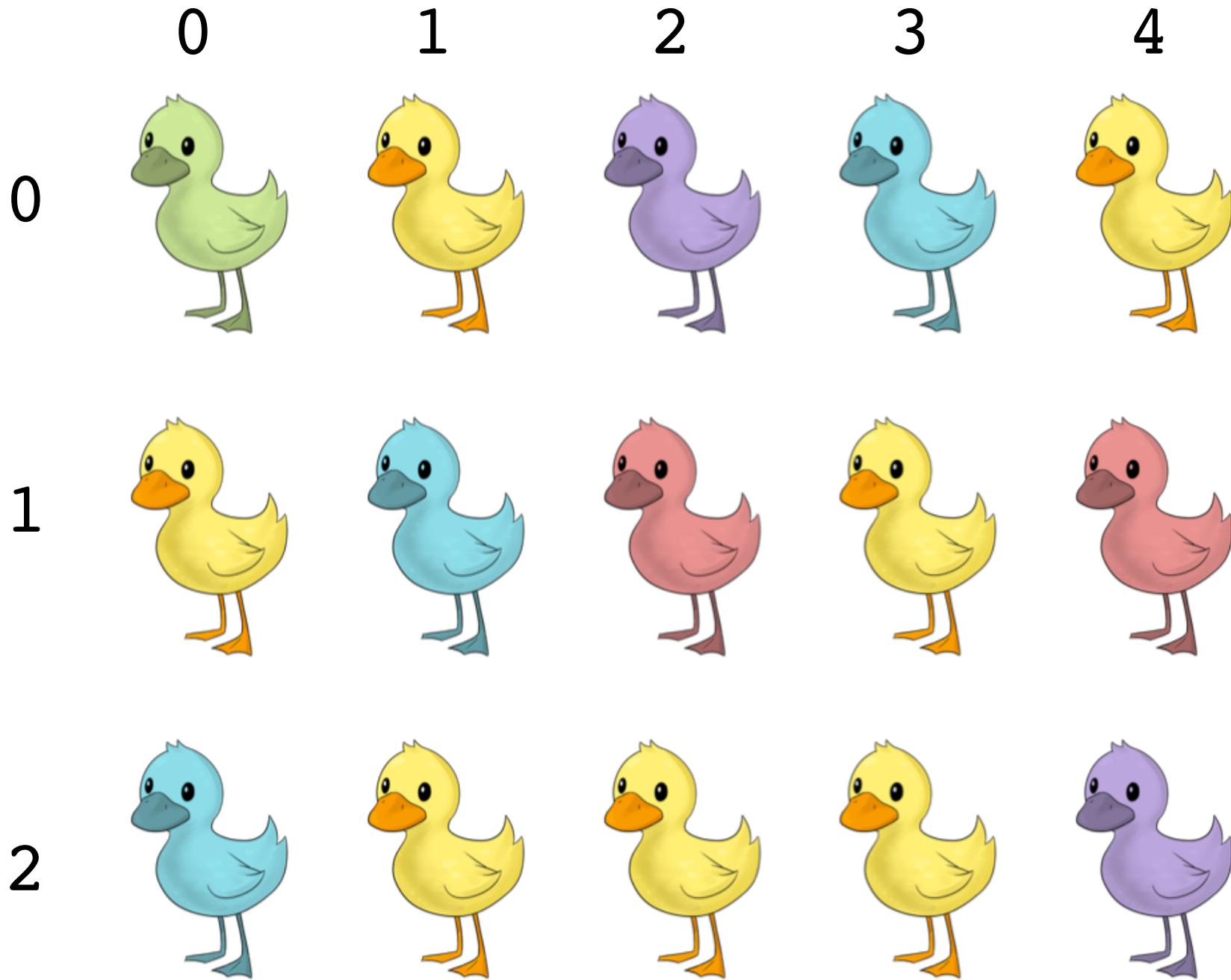


```
Duck[ ] duckArray = new Duck[ 4 ];
```



```
Duck[ ] duckArray = new Duck[ 4 ];
```





* Attack of the clone army of ducks! Hide your children!



The Matrix



The Matrix



WELCOME TO
THE MATRIX!!!!!!

a.k.a. 2D arrays



My First Matrix

```
int[][] morpheus = new int[2][4];
```



My First Matrix

```
int[][] morpheus = new int[2][4];
```



My First Matrix

```
int[][] morpheus = new int[2][4];
```



My First Matrix

```
int[][] morpheus = new int[2][4];
```



My First Matrix

```
int[][] morpheus = new int[2][4];
```



My First Matrix

```
int[][][] morpheus = new int[2][4];
```

Number of cols
Number of rows



My First Matrix

```
int[][] morpheus = new int[2][4];
```

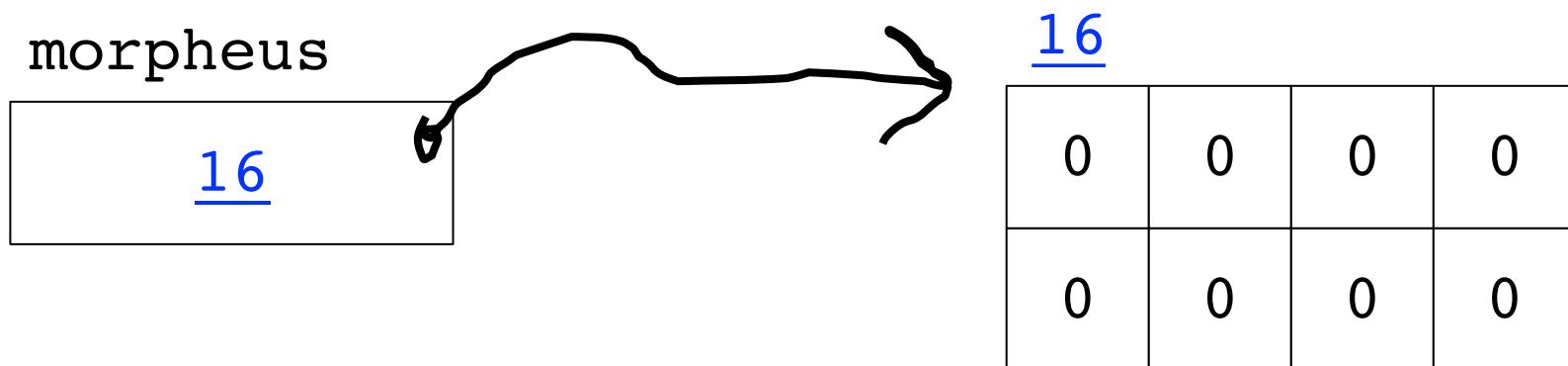
morpheus

0	0	0	0
0	0	0	0



My First Matrix

```
int[][] morpheus = new int[2][4];
```

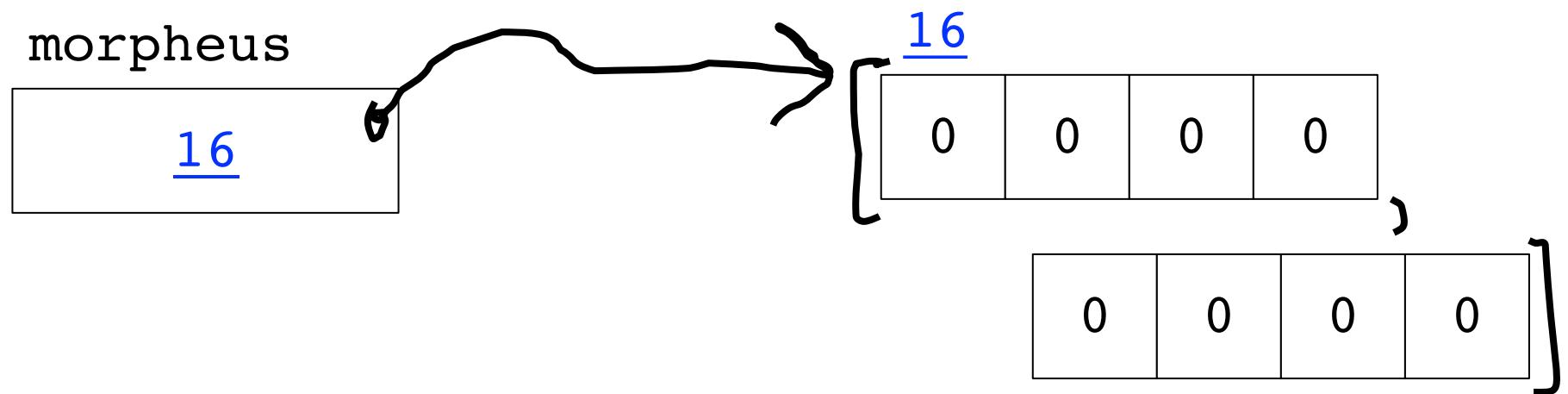


Aside: It's actually more like this.



My First Matrix

```
int[][] morpheus = new int[2][4];
```



If we are going to be brutally honest



My First Matrix

```
int[][] morpheus = new int[2][4];
```

morpheus

0	0	0	0
0	0	0	0



My First Matrix

```
int[][][] morpheus = new int[2][4];
```

morpheus

0	0	0	0
0	0	0	0

Task: Make this cell hold the value 1



```
int[][][] morpheus = new int[2][4];
```

	Col 0	Col 1	Col 2	Col 3
Row 0	0	0	0	1
Row 1	0	0	0	0

Task: Make this cell hold the value 1



My First Matrix

```
morpheus[0][3] = 1;
```

Row 0

morpheus

Col 3

0	0	0	0
0	0	0	0



My First Matrix

```
morpheus[0][3] = 1;
```

morpheus

0	0	0	1
0	0	0	0



When “indexing” into a matrix,
row comes first, then column.



```
myMatrix[      row      ] [      col      ]
```



When “indexing” into a matrix,
row comes first, then column.



R is for stanfoRd

myMatrix[



C is for Cal



When “indexing” into a matrix,
row comes first, then column.



Matrix: The revolutions

Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```

Before the method call:

matrix

0	0	4	0
0	17	0	0
0	0	0	6



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```

After the method call:

matrix

1	1	1	1
1	1	1	1
1	1	1	1



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}  
  
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}  
  
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

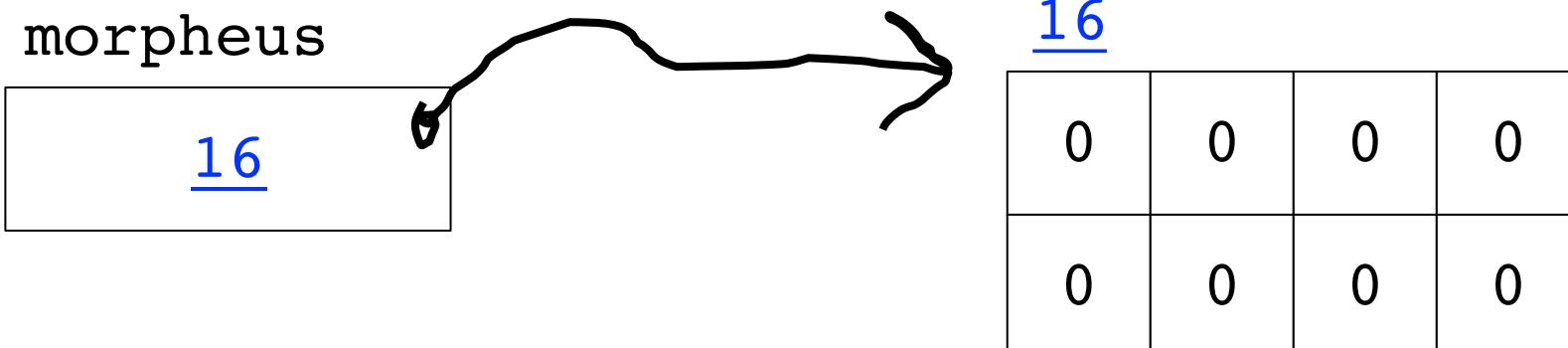
```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}  
  
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```

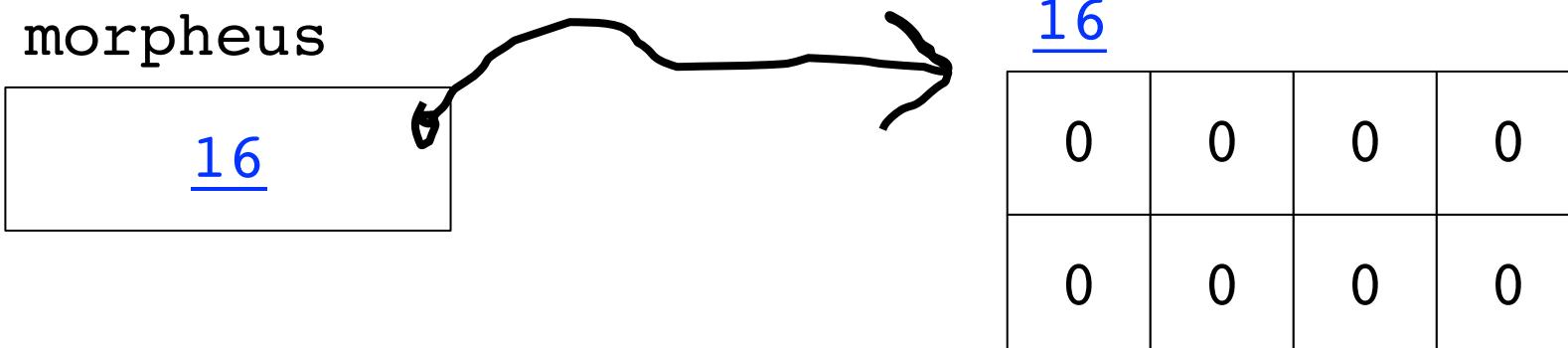
```
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```

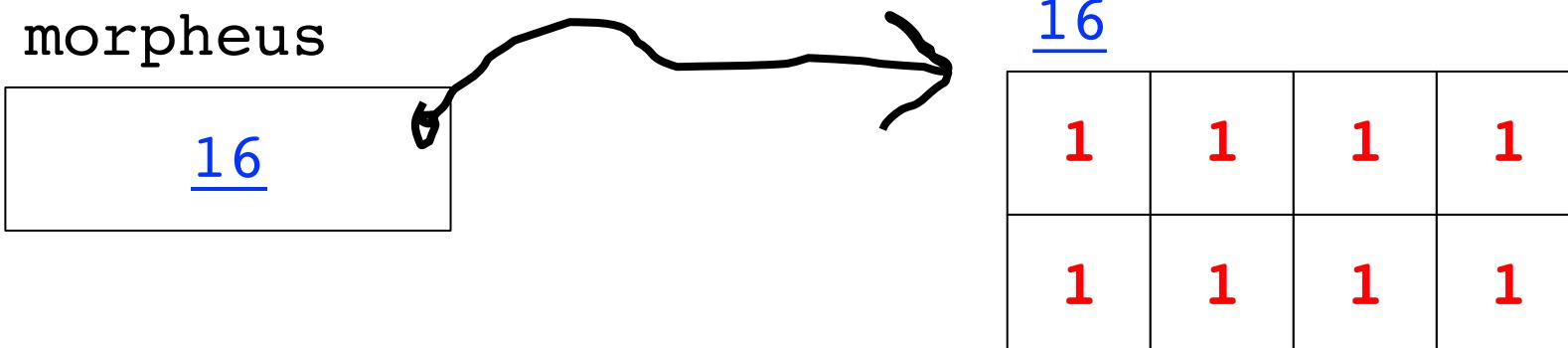
```
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```

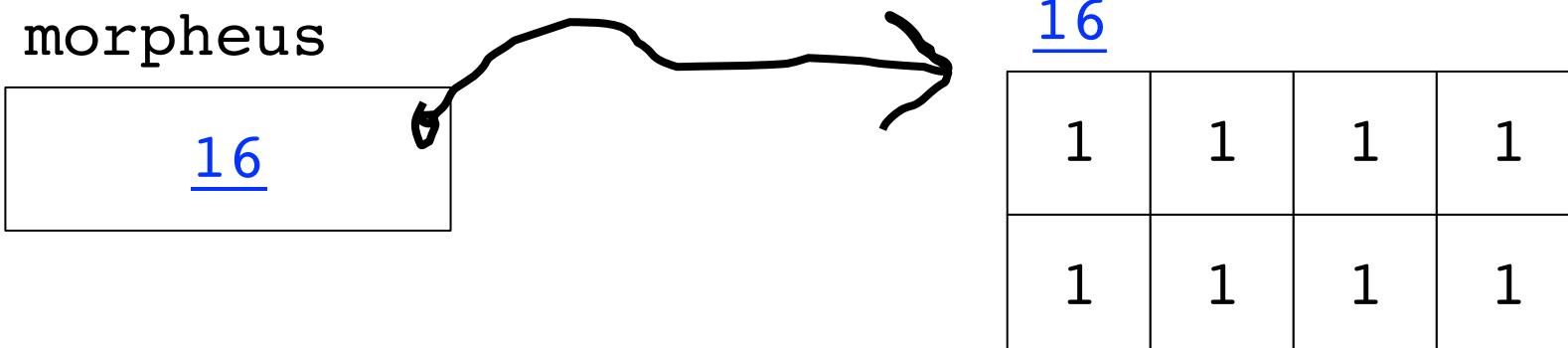
```
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```

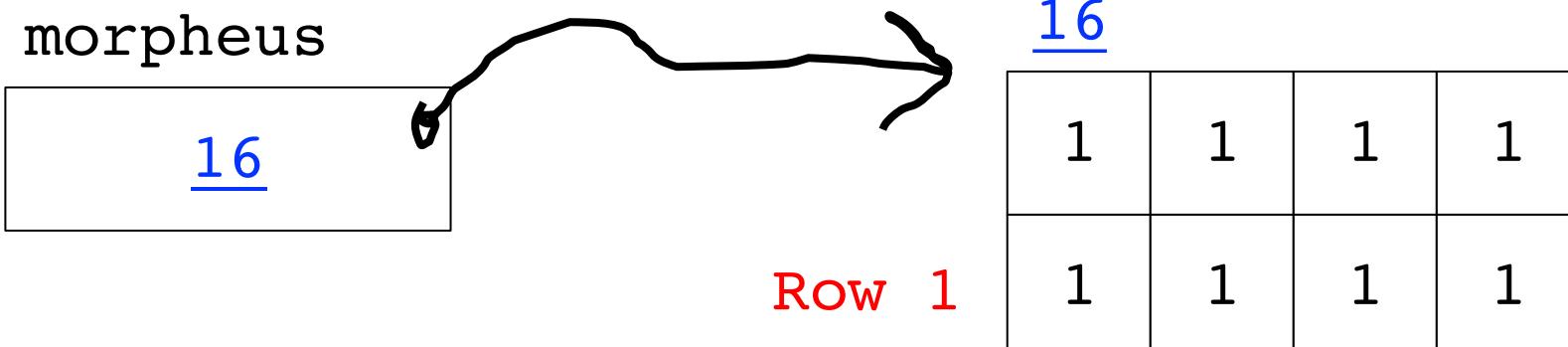
```
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```

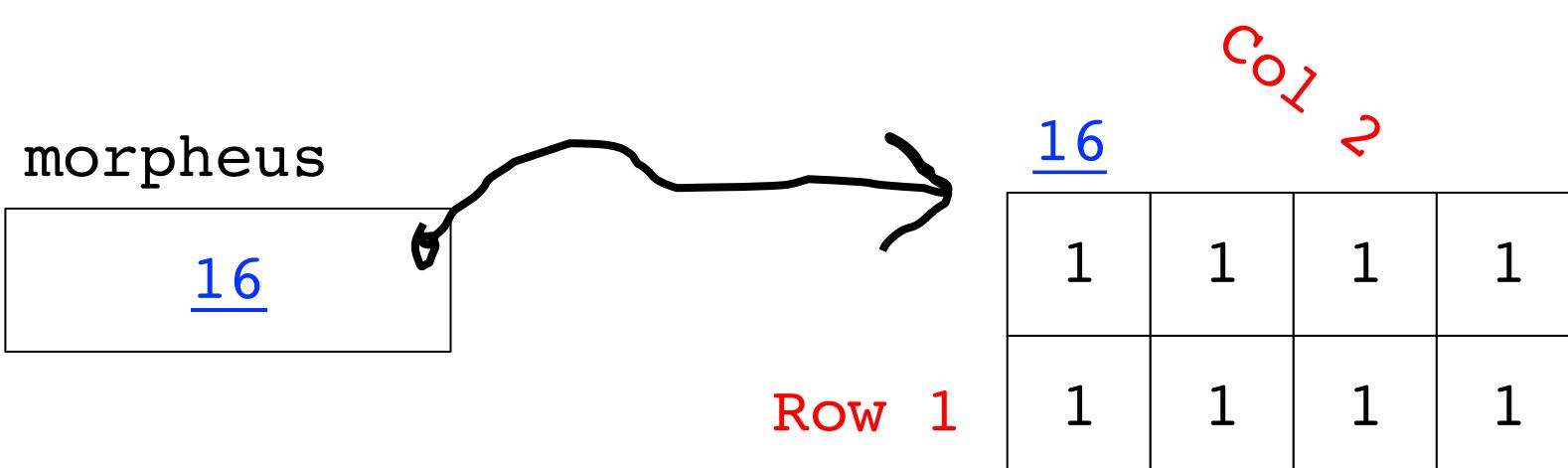
```
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```

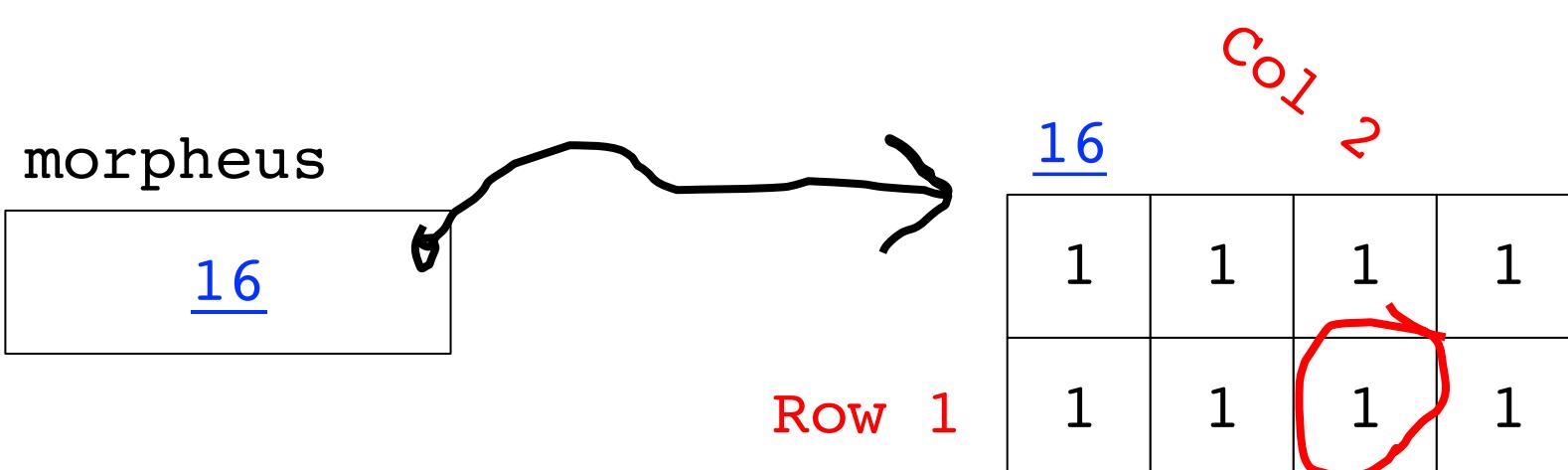
```
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```

```
public void run(){  
    int[][][] morpheus = int[4][2];  
    setValuesToOne(morpheus);  
    println(morpheus[1][2]);  
}
```



Set Values to One

```
private void setValuesToOne(int[][][] matrix) {  
    // your code here...  
}
```



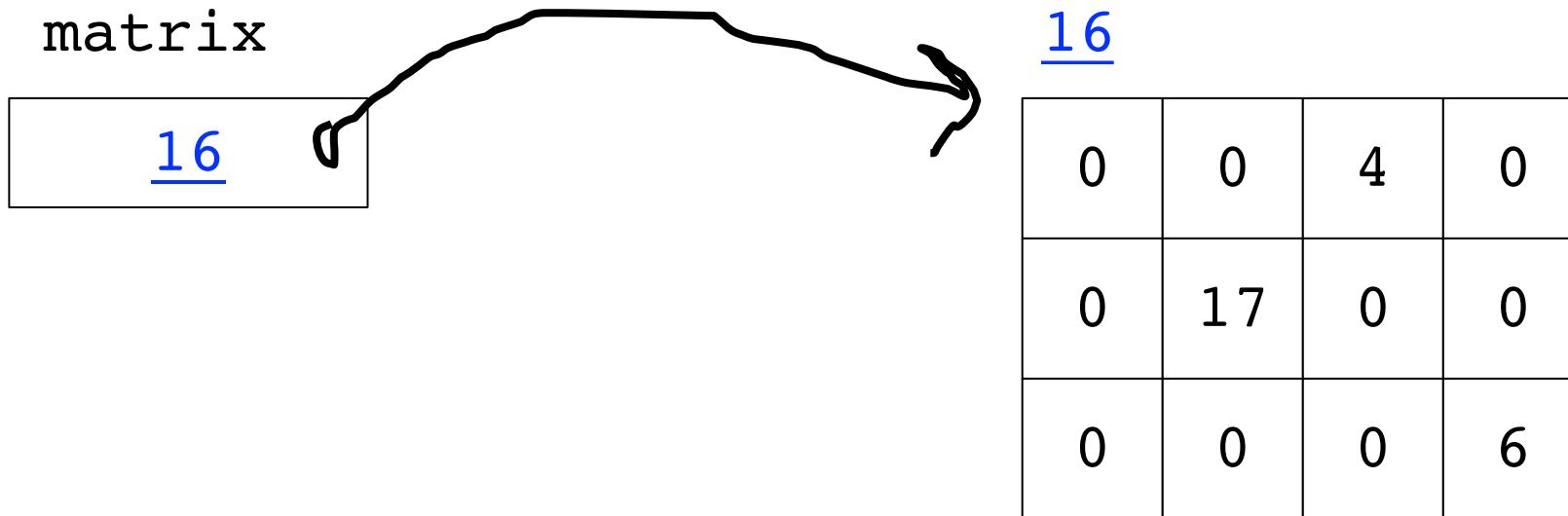
Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    // your code here...  
}
```



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    // your code here...  
}
```



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    // your code here...  
}
```

matrix

0	0	4	0
0	17	0	0
0	0	0	6



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    for(each row r) {  
        for(each col c) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```

matrix

0	0	4	0
0	17	0	0
0	0	0	6



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    for(int r = 0; r < 3; r++) {  
        for(int c = 0; c < 4; c++) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```

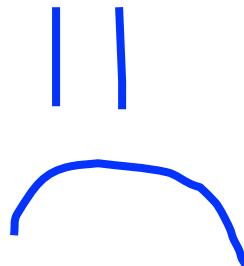
matrix

0	0	4	0
0	17	0	0
0	0	0	6



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    for(int r = 0; r < 3; r++) {  
        for(int c = 0; c < 4; c++) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```



matrix

0	0	4	0	4	0
0	17	0	0	0	0



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    for(int r = 0; r < matrix.length; r++) {  
        for(int c = 0; c < matrix[0].length; c++) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```

Works for
this matrix

matrix

0	0	4	0	4	0
0	17	0	0	0	0



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    for(int r = 0; r < matrix.length; r++) {  
        for(int c = 0; c < matrix[0].length; c++) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```

Also works for
this matrix

matrix

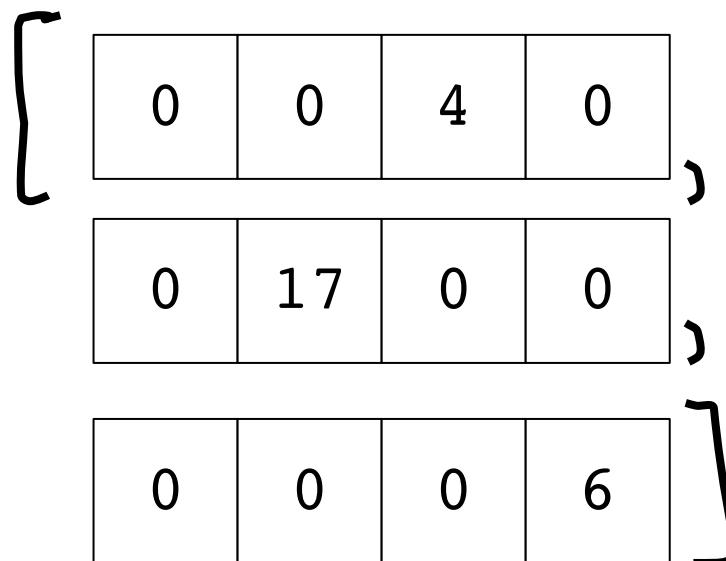
0	0	4	0
0	17	0	0
0	0	0	6



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    for(int r = 0; r < matrix.length; r++) {  
        for(int c = 0; c < matrix[0].length; c++) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```

matrix



Set Values to One

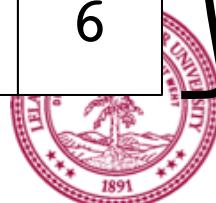
```
private void setValuesToOne(int[][] matrix) {  
    for(int r = 0; r < matrix.length; r++) {  
        for(int c = 0; c < matrix[0].length; c++) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```

matrix

0	0	4	0
---	---	---	---

0	17	0	0
---	----	---	---

0	0	0	6
---	---	---	---



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    for(int r = 0; r < matrix.length; r++) {  
        for(int c = 0; c < matrix[0].length; c++) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```

matrix

0	0	4	0
---	---	---	---

0

0	17	0	0
---	----	---	---

1

0	0	0	6
---	---	---	---

2



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    for(int r = 0; r < matrix.length; r++) {  
        for(int c = 0; c < matrix[0].length; c++) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```

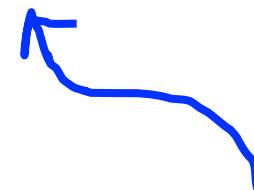
matrix

0	0	4	0
0	17	0	0
0	0	0	6



Set Values to One

```
private void setValuesToOne(int[][] matrix) {  
    for(int r = 0; r < numRows(matrix); r++) {  
        for(int c = 0; c < numCols(matrix); c++) {  
            matrix[r][c] = 1;  
        }  
    }  
}
```



These aren't defined.
But I highly recommend
them :-)

matrix

0	0	4	0
0	17	0	0
0	0	0	6



How to get number of rows

```
private int numRows(int[][] matrix) {  
    return matrix.length;  
}
```



How to get number of cols

```
private int numCols(int[][] matrix) {  
    return matrix[0].length;  
}
```



2D Arrays on one slide

1. Make a Matrix

```
double[][] mahMatrix = new double[nRows][nCols];
```

2. Set and get values from a matrix using bracket notation

```
mahMatrix[4][2] = 99.0;  
println(mahMatrix[0][0]);
```

3. Get the number of rows and columns of a matrix (pro-tip: define method)

```
int nRows = mahMatrix.length;  
int nCols = mahMatrix[0].length
```

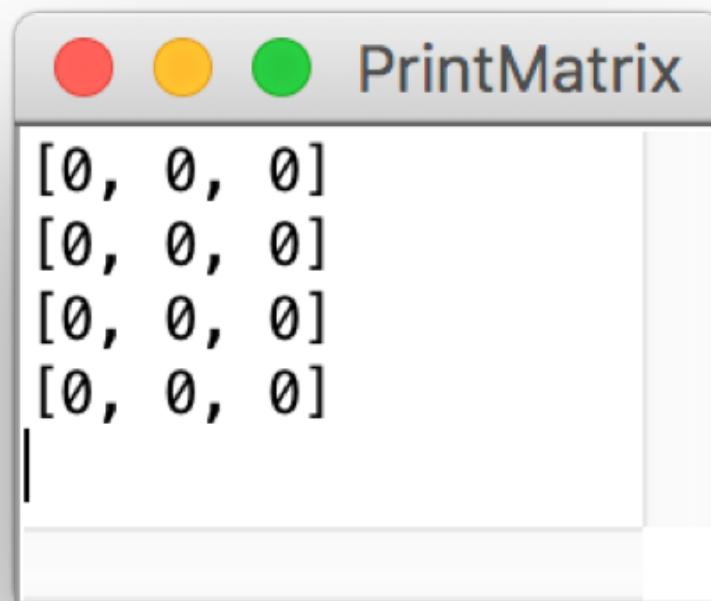
4. Use a double for loop to iterate over the whole matrix

```
for(int r = 0; r < mahMatrix.length; r++) {  
    for(int c = 0; c < mahMatrix[0].length; c++) {  
        //party pixel: mahMatrix[r][c]  
    }  
}
```



Your turn: print matrix

```
private void printMatrix(int[][] matrix){  
    // TODO: fill this in  
}
```



A deadly lack of style...

```
private void setValuesToTwo(int[][] matrix) {  
    for(int i = 0; i < matrix[0].length; i++) {  
        for(int j = 0; j < matrix.length; j++) {  
            matrix[i][j] = 2;  
        }  
    }  
}
```



A deadly lack of style...

```
private void setValuesToTwo(int[][][] matrix) {  
    for(int i = 0; i < matrix[0].length; i++) {  
        for(int j = 0; j < matrix.length; j++) {  
            matrix[i][j] = 2;  
        }  
    }  
}
```

A deadly lack of style...

```
private void setValuesToTwo(int[][] matrix) {  
    for(int i = 0; i < matrix[0].length; i++) {  
        for(int j = 0; j < matrix.length; j++) {  
            matrix[i][j] = 2;  
        }  
    }  
}
```



A deadly lack of style...

```
private void setValuesToTwo(int[][] matrix) {  
    for(int i = 0; i < matrix[0].length; i++) {  
        for(int j = 0; j < matrix.length; j++) {  
            matrix[i][j] = 2;  
        }  
    }  
}
```



A deadly lack of style...

```
private void setValuesToTwo(int[][] matrix) {  
    for(int r = 0; r < matrix[0].length; r++) {  
        for(int j = 0; j < matrix.length; j++) {  
            matrix[r][j] = 2;  
        }  
    }  
}
```



A deadly lack of style...

```
private void setValuesToTwo(int[][] matrix) {  
    for(int r = 0; r < matrix[0].length; r++) {  
        for(int c = 0; c < matrix.length; c++) {  
            matrix[r][c] = 2;  
        }  
    }  
}
```



A deadly lack of style...

```
private void setValuesToTwo(int[][] matrix) {  
    for(int r = 0; r < numRows(matrix); r++) {  
        for(int c = 0; c < numCols(matrix); c++) {  
            matrix[r][c] = 2;  
        }  
    }  
}
```



Images are Matrices!



Images are Matrices!

```
GImage img = new GImage("snowman.jpg");  
int[][] pixels = img.getPixelArray();
```

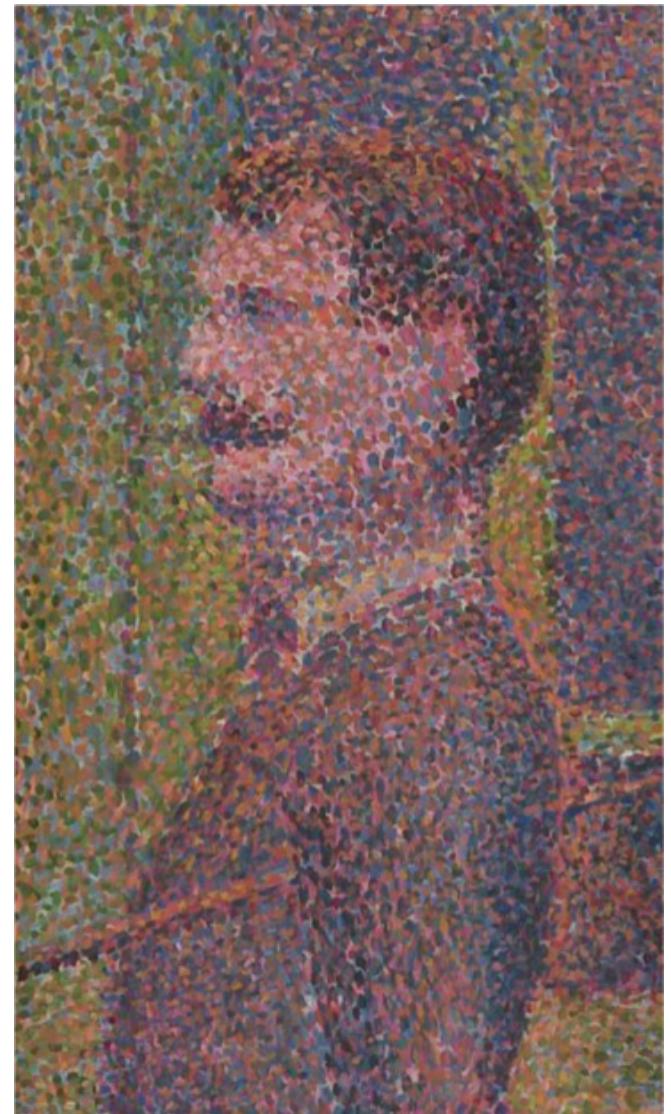
Creates an image
from a matrix of
pixels

Gives you the image
as a matrix of ints
(which you can edit)

```
GImage copy = new GImage(pixels);
```



Part two: Surat meets Instagram



Seurat: French post impressionist painter



Pointillism Filter

Repeat many times:

1. Pick a random pixel from an image.
2. Find the pixel's color
3. "Paint" a rather large brush stroke at a corresponding location, with the color





PICUH, CS100A, Stanford University





Piech, CS106A, Stanford University



c = 36

r = 24



c = 36

r = 24



c = 21

r = 38



c = 21

r = 38

