**Topic 1: 2D Arrays (40 points)**

1. Watch the videos in Topic 1: 2D Arrays.
2. Answer the following questions in complete sentences in your Word document:

* What is a 2D array? (3 points)
  + A 2D array is an array which stores other arrays in a grid pattern. Each row stores a column which is a 1D array, similar to a matrix.
* Draw or create a visual representation of a 2D string array that matches this line of code. Use words to represent the strings in the elements. String[][] grid = new String[2][4]; (5 points)
  + {[0], [0]
  + [0], [0]
  + [0], [0]
  + [0], [0]}
* Write a line of code that will declare and initialize a 2D array for a seating chart that has five rows of desks for 30 students. (4 points)
  + String[][] seatingMatrix = new String[5][6];
* Write a line of code to place Taylor at desk 5 in row 4. (2 points)
  + seatingMatrix[3][4] = “Taylor”;
* Write a line of code that accesses the last student in the last row using the length() method. (2 points)
  + seatingMatrix[seatingMatrix.length() - 1][seatingMatrix[0].length() - 1]
* Taylor has requested a seating change closer to the board. Cheyanne's seat is the third seat in the first row. Write the code needed to swap Taylor and Cheyanne. (4 points)
  + seatingMatrix[0][2] = “Taylor”
  + seatingMatrix[3][4] = “Cheyanne”

**Topic 2: Traversing 2D Arrays (60 points)**

1. Watch the videos in Topic 2: Traversing 2D Arrays.
2. Answer the following questions in complete sentences in your Word document:

* Write a void method using the length() method to print out your String 2D array from question 2 from Topic 1. (5 points)
  + public static void printMatrix(String matrix[][]) {
  + for (int row = 0; row < matrix.length; row++) {
  + for (int column = 0; column < matrix[row].length; column++) {
  + System.out.println(matrix[row][column]);
  + }
  + }
  + }
* Rewrite your void method using enhanced for loops. (5 points)
  + public static void printEnhancedMatrix(String matrix[][]) {
  + for (String[] row : matrix) {
  + for (String column : row) {
  + System.out.println(column);
  + }
  + }
  + }
* Rewrite your void method in column-major order. (5 points)
  + public static void printColumnMajorMatrix(String matrix[][]) {
  + for (int column = 0; column < matrix[0].length; column++) {
  + for (int row = 0; row < matrix.length; row++) {
  + System.out.println(matrix[row][column]);
  + }
  + }
  + }
* Write a boolean search method for the seating chart 2D array in Topic 1. (5 points)
  + public static boolean searchMatrix(String matrix[][], String key) {
  + for (int row = 0; row < matrix.length; row++) {
  + for (int column = 0; column < matrix[row].length; column++) {
  + String current = matrix[row][column];
  + if (current != null && current.equals(key)) {
  + return true;
  + }
  + }
  + }
  + return false;
  + }
* Write a method called findLongest() which will return the longest string in your String array from Topic 1. (5 points)
  + public static String findLongest(String matrix[][]) {
  + String longest = "";
  + for (int row = 0; row < matrix.length; row++) {
  + for (int column = 0; column < matrix[row].length; column++) {
  + if (matrix[row][column] == null) {
  + continue;
  + }
  + String current = matrix[row][column];
  + if (current.length() > longest.length()) {
  + longest = current;
  + }
  + }
  + }
  + return longest;
  + }
  + }
* In video 8.2:Daily Video 3, the presenter said "that's kinda gauche, right?". Explain why he said that as well as the line of code where the quote was made. (5 points)
  + The presenter said “that’s kinda gauche, right” on the line “ if (matrix[row][column]” because the matrix is already a matrix of booleans, so there is no need to compare a boolean in an if statement because an if statement already takes in a boolean value. By specifying “matrix[row][column]”, the programmer avoids writing unnecessary code.

All of my code in godbolt (commit to github later):

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online Java Compiler.

Code, Compile, Run and Debug java program online.

Write your code in this editor and press "Run" button to execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

public class Main

{

public static void main(String[] args) {

String[][] boogers = new String[10][5];

boogers[2][1] = "69420";

boogers[6][4] = "hi";

printColumnMajorMatrix(boogers);

System.out.println(searchMatrix(boogers, "69420"));

System.out.println(findLongest(boogers));

}

public static void printMatrix(String matrix[][]) {

for (int row = 0; row < matrix.length; row++) {

for (int column = 0; column < matrix[row].length; column++) {

System.out.println(matrix[row][column]);

}

}

}

public static void printEnhancedMatrix(String matrix[][]) {

for (String[] row : matrix) {

for (String column : row) {

System.out.println(column);

}

}

}

public static void printColumnMajorMatrix(String matrix[][]) {

for (int column = 0; column < matrix[0].length; column++) {

for (int row = 0; row < matrix.length; row++) {

System.out.println(matrix[row][column]);

}

}

}

public static boolean searchMatrix(String matrix[][], String key) {

for (int row = 0; row < matrix.length; row++) {

for (int column = 0; column < matrix[row].length; column++) {

String current = matrix[row][column];

if (current != null && current.equals(key)) {

return true;

}

}

}

return false;

}

public static String findLongest(String matrix[][]) {

String longest = "";

for (int row = 0; row < matrix.length; row++) {

for (int column = 0; column < matrix[row].length; column++) {

if (matrix[row][column] == null) {

continue;

}

String current = matrix[row][column];

if (current.length() > longest.length()) {

longest = current;

}

}

}

return longest;

}

}