Name: Naga Venkatesh Gavini
NEU ID: 002705198

CSYE 6200 Assignment-4

Problem 1:

Problem Description:
Write a program to accept the number of rows and columns from the user. Once the user enters the rows and columns, create a matrix and fill it with the numbers randomly from 0-9. After that, calculate the row which has a largest sum and column with the largest sum. Print the row and column indexes which has the largest sum. To generate random numbers, refer to the Random class of Java.

Analysis:

Algorithm:

- Firstly, the program will ask the user to input the rows and column values for the construction of the 2D array.
- Once the input is provided by the user, program start using the Random module in java to create the 2D array according to the rows and column values.
- Next step is to find the largest row with highest sum, for this we use a for loop to iterate through each row and calculate the largest row sum by keeping a separate variable. This variable gets updated as soon as we find a new sum.
    - The following function is going to be used to calculate the sum for the row.
    - *sumRow(int row[])* – This method takes an array as a parameter, it iterates through array elements and performs addition on them further returns the sum of the entire array.
- Similar process will be followed for column as well. In this step, we send the matrix and the column number to the function *sumColumn***,** it will return the sum value of the column.
    - The following function will be used to calculate the sum for the column.
    - *sumColumn(int matrix[][], int column)* – This method takes a 2 D array(matrix) as well as column index, it iterates through matrix and along the column index, adds its elements, returns the sum of entire column.

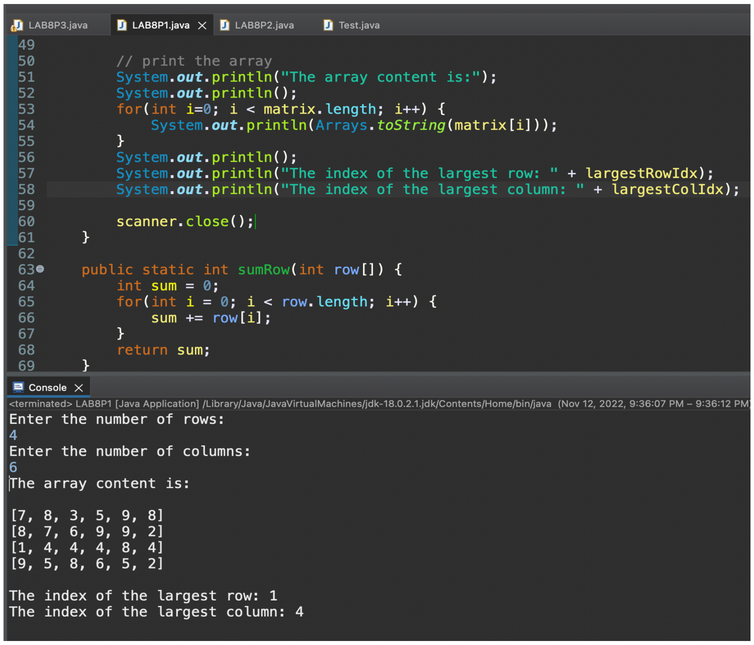
Difficulties Faced:
Didn’t faced many challenges while doing this problem, instead I got to learn about the multi-dimensional arrays and their usage in Java.

Source Code for 1st problem:

```java
package edu.northeastern.csye6200;

import java.util.Arrays;
import java.util.Random;
import java.util.Scanner;

public class LAB8P1 {
  public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the number of rows: ");
            int rows = scanner.nextInt();
            System.out.println("Enter the number of columns: ");
            int cols = scanner.nextInt();
            Random rndRandom = new Random();


            int[][] matrix = new int[rows][cols];

            // Fill the array
            for(int i = 0; i < rows; i++) {
                      for(int j = 0; j < cols; j++) {
                                matrix[i][j] = rndRandom.nextInt(10);
                      }
            }

            // Check for largest row
            int largestRowIdx = -1;
            int rowSumPlaceHolder = 0;
            int newSum = 0;
            for(int i = 0; i < rows; i++) {
                      newSum = sumRow(matrix[i]);
                      if(newSum > rowSumPlaceHolder) {
                                largestRowIdx = i;
                                rowSumPlaceHolder = newSum;
                      }
            }

            // Check for largest column
            int largestColIdx = -1;
            int colSumPlaceholder = 0;
            int newSumCol = 0;
            for(int i = 0; i < cols; i++) {
                      newSumCol = sumColumn(matrix, i);
                      if(newSumCol > colSumPlaceholder) {
                                largestColIdx = i;
                                colSumPlaceholder = newSumCol;
                      }
            }

            // print the array
            System.out.println("The array content is:");
            System.out.println();
            for(int i=0; i < matrix.length; i++) {
                      System.out.println(Arrays.toString(matrix[i]));
            }
            System.out.println();
            System.out.println("The index of the largest row: " + largestRowIdx);
            System.out.println("The index of the largest column: " + largestColIdx);
  }

```

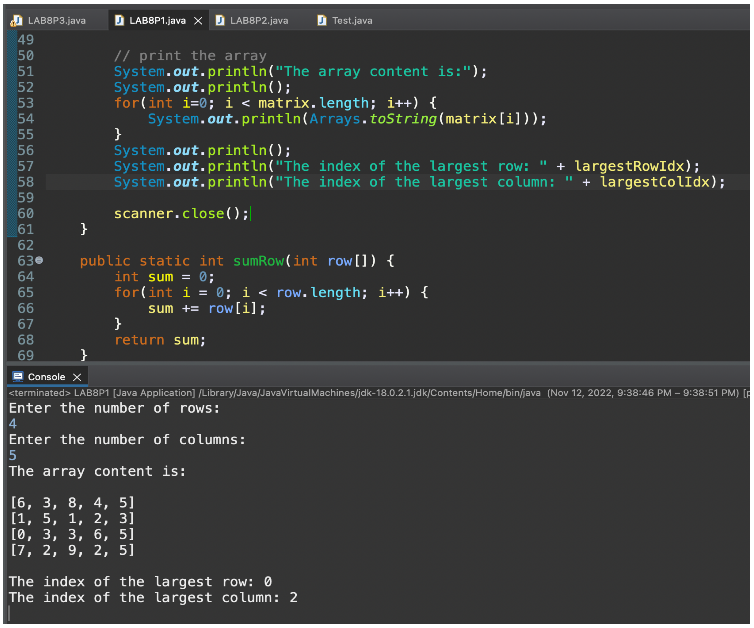
```
    public static int sumRow(int row[]) {
            int sum = 0;
            for(int i = 0; i < row.length; i++) {
                    sum += row[i];
            }
            return sum;
    }

    public static int sumColumn(int matrix[][], int column) {
            int sum = 0;
            for(int i = 0; i < matrix.length; i++) {
                    sum += matrix[i][column];
            }

            return sum;
    }
  }

```

Screenshots for 1st problem:



```

        // print the array
        System.out.println("The array content is:");
        System.out.println();
        for(int i=0; i < matrix.length; i++) {
            System.out.println(Arrays.toString(matrix[i]));
        }
        System.out.println();
        System.out.println("The index of the largest row: " + largestRowIdx);
        System.out.println("The index of the largest column: " + largestColIdx);

        scanner.close();
    }

    public static int sumRow(int row[]) {
        int sum = 0;
        for(int i = 0; i < row.length; i++) {
            sum += row[i];
        }
        return sum;
    }
```

```
<terminated> LAB8P1 [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java  (Nov 12, 2022, 9:36:07 PM – 9:36:12 PM)
Enter the number of rows:
4
Enter the number of columns:
6
The array content is:

[7, 8, 3, 5, 9, 8]
[8, 7, 6, 9, 9, 2]
[1, 4, 4, 4, 8, 4]
[9, 5, 8, 6, 5, 2]

The index of the largest row: 1
The index of the largest column: 4
```



```java
        // print the array
        System.out.println("The array content is:");
        System.out.println();
        for(int i=0; i < matrix.length; i++) {
            System.out.println(Arrays.toString(matrix[i]));
        }
        System.out.println();
        System.out.println("The index of the largest row: " + largestRowIdx);
        System.out.println("The index of the largest column: " + largestColIdx);

        scanner.close();
    }

    public static int sumRow(int row[]) {
        int sum = 0;
        for(int i = 0; i < row.length; i++) {
            sum += row[i];
        }
        return sum;
    }
```

```
<terminated> LAB8P1 [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java  (Nov 12, 2022, 9:38:46 PM – 9:38:51 PM) [p
Enter the number of rows:
4
Enter the number of columns:
5
The array content is:

[6, 3, 8, 4, 5]
[1, 5, 1, 2, 3]
[0, 3, 3, 6, 5]
[7, 2, 9, 2, 5]

The index of the largest row: 0
The index of the largest column: 2
```

Problem2:

Problem Description:

Program to display the bar chart for the percentages of the overall grades. Total grade of the class will be represented by project, exams, assignments and attendance. Suppose project takes 35 percent and it is displayed by blue. Exams take 30 percent, and it is displayed in green. Assignments take 30 percent and are displayed in red color. Attendance takes 5 percent and is displayed in orange color. Use the JavaFX rectangle class for this problem to display the chart as mentioned.

Analysis:

Algorithm:

- To solve this problem, we are storing the constants to store the height, weight and the percentage fractions for the final bar graphs.
- Along with the above variables, we are storing colors such as blue, green, orange according to the problem requirements.
- *public void start(Stage primaryStage)* – Pane class has been used to insert the proper layout the final output which includes bar graphs, height of all the plots, title of the plots. The Scene which is a container for all visual components is set on a stage container using the setScene() method. The setTitle() and show() methods are used to set title for stage and display scene contents respectively.

- Main method is used to launch the application using launch method, which will internally invokes the start() method.

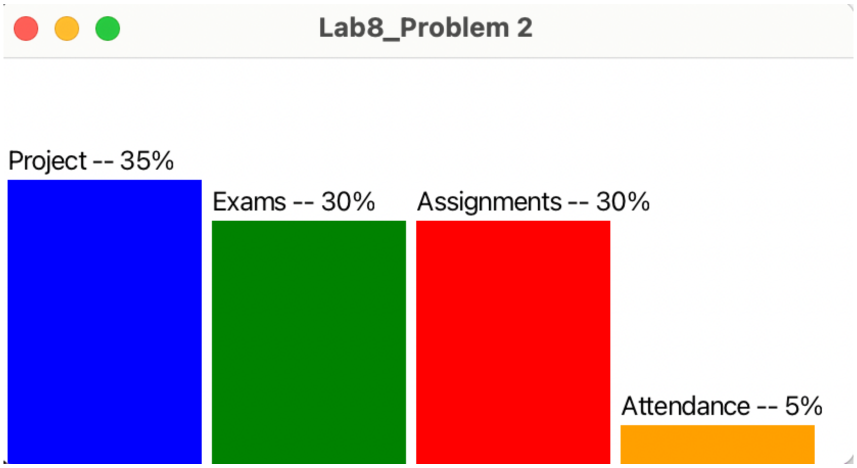
Difficulties Faced:

- Faced difficulty on deciding the x and y coordinated for the bar graphs.
- Faced difficulty to figure out an optimal way to construct the required output.
- Spent some time to understand the Grid pane and Scene classes in JavaFx.

Source Code for 2nd problem:

```
package edu.northeastern.csye6200;

import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Rectangle;
import javafx.scene.text.Text;
import javafx.stage.Stage;


public class LAB8P2 extends Application
{
  public static final double WIDTH = 400;
  public static final double HEIGHT = 400;
  public static final double HALFHEIGHT = (HEIGHT / 2);

  public static double[] percenVals = {0.35, 0.30, 0.30, 0.05};

  public static  Color[] allColors = {
                      Color.BLUE,
                      Color.GREEN,
                      Color.RED,
                      Color.ORANGE
            };

  public static String[] barTitles = {
            "Project -- " + String.format("%d%s", (int)(100 * percenVals[0]), "%"),
            "Exams -- " + String.format("%d%s", (int)(100 * percenVals[1]), "%"),
            "Assignments -- " + String.format("%d%s", (int)(100 * percenVals[2]), "%"),
            "Attendance -- "+ String.format("%d%s", (int)(100 * percenVals[3]), "%")
            };


  @Override
  public void start(Stage primaryStage){
            Pane pane = new Pane();
            pane.setPadding(new Insets(5, 10, 0, 10));
            Rectangle[] bars = new Rectangle[4];

            for (int i = 0; i < bars.length; i++){
                      // Create rectangle
                      bars[i] = new Rectangle(
                                          5 + (100 * i),
                                          HALFHEIGHT - (HEIGHT * percenVals[i]),
                                          WIDTH / bars.length - 5,
                                          HEIGHT * percenVals[i]
                      );

                      // Fill with color
                      bars[i].setFill(allColors[i]);

                      // Set the title for the bar
                      Text mytext = new Text(
                                         5 + (100 * i),
                                         HALFHEIGHT - (HEIGHT * percenVals[i]) - 5,
                                         barTitles[i]
                      );

                      pane.getChildren().addAll(mytext, bars[i]);
            }

            primaryStage.setScene(new Scene(pane, WIDTH + 20, HALFHEIGHT));
            primaryStage.setTitle("Lab8_Problem 2");
            primaryStage.show();
  }

  public static void main(String[] args){
            launch(args);
  }
}

```

Screenshot for 2nd problem:



Problem3:

Problem Description:

Create and implement a software that shows a 10 × 10 binary value matrix, or 0s and 1s. The setText() method must be used to display the numbers in text fields. Use Random class in Java to generate randomness.

Analysis:

Algorithm:

- The matrix's dimensions are specified in the start() function, and the grid's display layout is constructed with the help of Grid Pane class of Java.
- With each loop iteration over the matrix, a new text field is created, into which the randomly generated numbers are entered. The scene has been set.
- At each iteration, the Random class was used to generate random numbers between 0 and 1 and add it to the text Field object.
- Main method is used to launch the application using launch method, which will internally invokes the start() method.

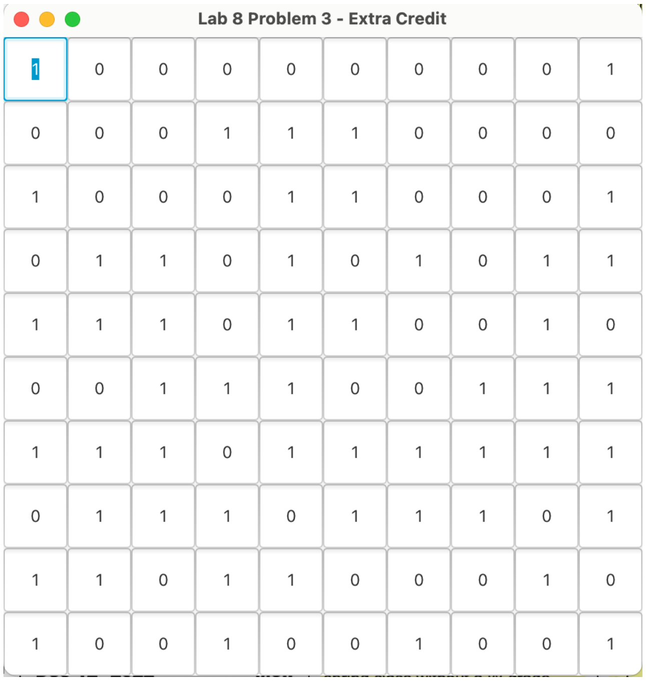
Difficulties Faced:

Didn’t faced many difficulties during this problem since I already went through the grid pane and Random class concepts in the last problems.

Source Code for 3rd problem:

```java
package edu.northeastern.csye6200;

import java.util.Random;

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

public class LAB8P3 extends Application {

  @Override
  public void start(Stage primaryStage) {
        GridPane grids = new GridPane();

        for(int i = 0; i < 10; i++){
            for(int j = 0; j < 10; j++){

                Random rand = new Random();
                int rand1 = rand.nextInt(2);
                String randInString = String.valueOf(rand1);

                TextField textField = new TextField();
                textField.setPrefHeight(50);
                textField.setPrefWidth(50);
                textField.setAlignment(Pos.CENTER);
                textField.setEditable(false);
                textField.setText(randInString);

                grids.setRowIndex(textField, i);
                grids.setColumnIndex(textField, j);
                grids.getChildren().add(textField);
            }
        }

        Scene scene = new Scene(grids, 500, 500);
        primaryStage.setTitle("Lab 8 Problem 3 - Extra Credit");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
            launch(args);
    }

}

```

Screenshot for 3rd problem:



1	0	0	0	0	0	0	0	0	1
0	0	0	1	1	1	0	0	0	0
1	0	0	0	1	1	0	0	0	1
0	1	1	0	1	0	1	0	1	1
1	1	1	0	1	1	0	0	1	0
0	0	1	1	1	0	0	1	1	1
1	1	1	0	1	1	1	1	1	1
0	1	1	1	0	1	1	1	0	1
1	1	0	1	1	0	0	0	1	0
1	0	0	1	0	0	1	0	0	1