Distance Between 2 Cities Exercise

This program is going to find the distance between 2 cities by using a two-dimensional array to store the distances.

1. The name of this project and class should be **DistanceCities**.

2. Add your comments to the top of the program.

3. Add the JOptionPane import:

   import javax.swing.JOptionPane;

4. Let's determine the starting and ending city by asking the user to choose from a list of cities that we will calculate the distance between and store in our array.

```
String city1 = JOptionPane.showInputDialog(null,
        "What is your starting city: Dallas, Austin, Houston, "
                + "Galveston?");
String city2 = JOptionPane.showInputDialog(null,
        "What is your ending city: Dallas, Austin, Houston, "
                + "Galveston?");
```

5. The distances between the cities have been calculated for you in the following chart. These values will need to be entered into your code in the form of a 2-dimesional array.

	Dallas	**Austin**	**Houston**	**Galveston**
Dallas	0	195	239	289
Austin	195	0	145	189
Houston	239	145	0	52
Galveston	289	189	52	0

```
int distance[][] = {{0, 195, 239, 289},
                    {195, 0, 145, 189},
                    {239, 145, 0, 52},
                    {289, 189, 52, 0}};
```

6. Next, we will create a switch based on the city 1 that the user entered. We will use the city that they entered to determine which row of data we should pull the distance from. We will need to create a **row** variable to store the result of the switch once executed. The default case was added to the switch in case a user enters an invalid city.

		column 0	column 1	column 2	column 3
		Dallas	**Austin**	**Houston**	**Galveston**
row 0	**Dallas**	0	195	239	289
row 1	**Austin**	195	0	145	189
row 2	**Houston**	239	145	0	52
row 3	**Galveston**	289	189	52	0

```
int row = 0;
switch (city1) {
    case "Dallas":
        row = 0;
        break;
    case "Austin":
        row = 1;
        break;
    case "Houston":
        row = 2;
        break;
    case "Galveston":
        row = 3;
        break;
    default:
        row = 0;
        System.out.println("Invalid city");
        break;
}
```

7. Now that we have the row of the table to pull the data from, we need to know which column to pull the data from. Let's add a column variable to store the result of the switch statement for city 2. The default case was added to the switch in case a user enters an invalid city.

```
int column = 0;
switch (city2) {
    case "Dallas":
        column = 0;
        break;
    case "Austin":
        column = 1;
        break;
    case "Houston":
        column = 2;
        break;
    case "Galveston":
        column = 3;
        break;
    default:
        column = 0;
        System.out.println("Invalid city");
        break;
}
```

8. Finally, let's display the output box that will display the distances between our two cities. You will need to use the row and column number for the array since it is a two-dimensional array. The row was determined by the first city that the user entered and the column was determined by the second city that the user entered.

```
JOptionPane.showMessageDialog(null, "Distance is " +
        distance[row][column] + " miles");
```

9. Test your program to ensure that the distances are correct. Try entering Dallas as the first city and Dallas as the second city. What happens? Did you get 0 as the distance? Test all your cities.

10. The final version of the program should look as follows:

```
import javax.swing.JOptionPane;
public class DistanceCities {
    public static void main(String[] args) {
        String city1 = JOptionPane.showInputDialog(null,
                "What is your starting city: Dallas, Austin, Houston, "
                + "Galveston?");
        String city2 = JOptionPane.showInputDialog(null,
                "What is your ending city: Dallas, Austin, Houston, "
                + "Galveston?");
        int distance[][] = {{0, 195, 239, 289},
                            {195, 0, 145, 189},
                            {239, 145, 0, 52},
                            {289, 189, 52, 0}};
        int row = 0;
        switch (city1) {
            case "Dallas":
                row = 0;
                break;
            case "Austin":
                row = 1;
                break;
            case "Houston":
                row = 2;
                break;
            case "Galveston":
                row = 3;
                break;
            default:
                row = 0;
                System.out.println("Invalid city");
                break;
        }
        int column = 0;
        switch (city2) {
            case "Dallas":
                column = 0;
                break;
            case "Austin":
                column = 1;
                break;
            case "Houston":
                column = 2;
                break;
            case "Galveston":
                column = 3;
                break;
            default:
                column = 0;
                System.out.println("Invalid city");
                break;
        }
        JOptionPane.showMessageDialog(null, "Distance is " +
                distance[row][column] + " miles");
    }
}
```