Concept: Manipulating Arrays

Goal

Manipulate arrays using various LabVIEW functions.

Description

You are given a VI and asked to enhance it for a variety of purposes. For each part of this exercise, begin with the Array Investigation.vi located in the corresponding `<Exercise>` directory. The front panel of this VI is shown in Figure 1.

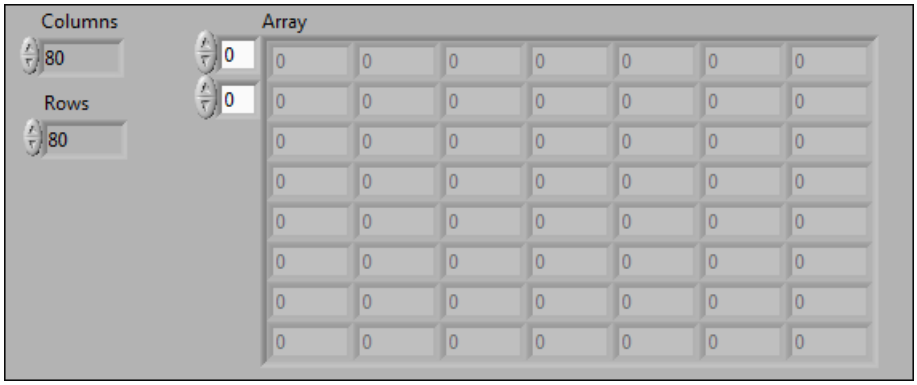

Figure 1. Array Investigation VI Front Panel

Figure 2 shows the block diagram of this VI.

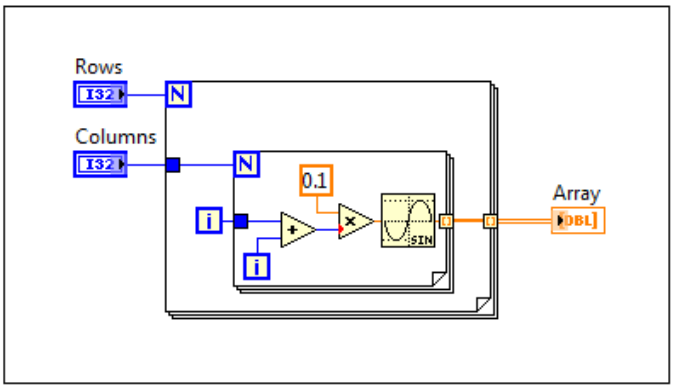

Figure 2. Array Investigation VI Block Diagram

This exercise is divided into two parts. You are given the scenario for each part first. Detailed implementation instructions for each part follow the scenario information.

Part 1: Iterate, Modify, and Graph Array

Modify the Array Investigation VI so that after the array is created, the array is indexed into For Loops where you multiply each element of the array by 100 and coerce each element to the nearest whole number. Graph the resulting 2D array to an intensity graph.

Part 2: Simplified Iterate, Modify, and Graph Array

Modify the Array Investigation VI or the solution from Part 1 to accomplish the same goals without using the nested For Loops.

Part 1: Implementation

Modify the Array Investigation VI so that after the array is created, the array is indexed into For Loops where you multiply each element of the array by 100 and coerce each element to the nearest whole number. Graph the resulting 2D array on an intensity graph.

The files that you need to complete this exercise are here:
`<NI eLearning>\LV Core 1\Array Grp Data\Exercise`.

1. Open `Array Investigation.vi` located in the `<Exercise>` directory.

2. Save the VI as `Array Investigation Part 1.vi`.

3. Add an intensity graph to the front panel of the VI and autoscale the X and Y axes, as shown in Figure 3. To verify that autoscaling is enabled for the axes, right-click the intensity graph and select **X Scale»AutoScale X** and **Y Scale»AutoScale Y** and ensure these items are checked.

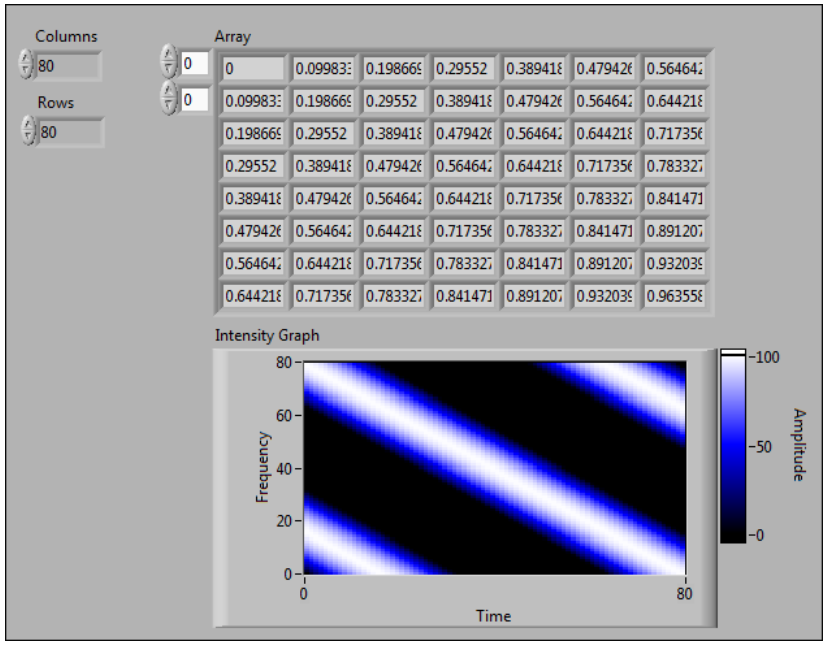

Figure 3. Array Investigation Part 1 VI Front Panel

4. Open the block diagram of the VI.

In the following steps, you create a block diagram similar to Figure 4.

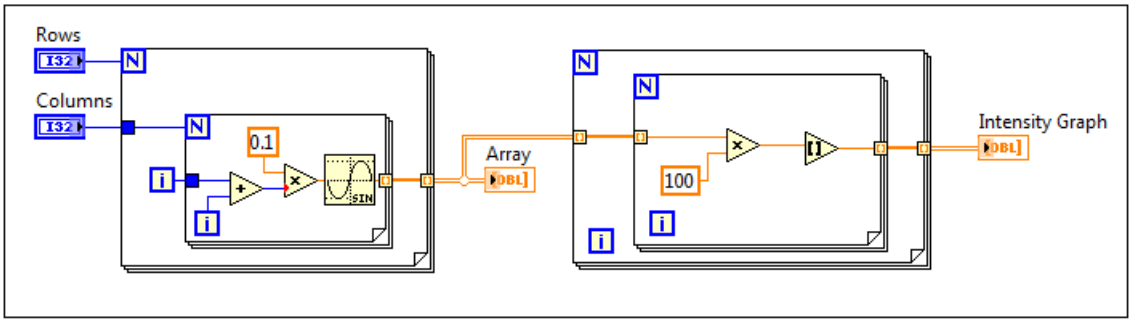

Figure 4. Array Investigation Part 1 VI Block Diagram

5. Iterate the Array.

   ❑ Add a For Loop to the right of the existing code.

   ❑ Add a second For Loop inside the first For Loop.

   ❑ Wire the array indicator terminal to the interior For Loop border. This creates an auto-indexed input tunnel on both For Loops.

6. Multiply each element of the array by 100.

   - ❑ Add a Multiply function to the interior For Loop.
   - ❑ Wire the indexed input tunnel to the x input of the Multiply function.
   - ❑ Right-click the y input and select **Create»Constant** from the shortcut menu.
   - ❑ Enter `100` in the constant.

7. Round each element to the nearest whole number.

   - ❑ Add a Round To Nearest function to the right of the Multiple function.
   - ❑ Wire the output of the Multiply function to the input of the Round To Nearest function.

8. Create a 2D array on the output of the For Loops to recreate the modified array.
   - ❑ Wire the output of the Round To Nearest function to the outer For Loop. This creates an auto-indexed output tunnel on both For Loops.
9. Wire the output array to the Intensity Graph indicator.
10. Switch to the front panel.
11. Save the VI.
12. Enter values for Rows and Columns.
13. Run the VI.

Part 2: Implementation

Modify Part 1 to accomplish the same goals without using the nested For Loops.

1. Open `Array Investigation Part 1.vi` if it is not still open.
2. Save the VI as `Array Investigation Part 2.vi`.
3. Open the block diagram.
4. Right-click the border of the interior For Loop, containing the Multiply function and the Round to Nearest function, and select **Remove For Loop**.

5. Right-click the border of the remaining For Loop and select **Remove For Loop** from the shortcut menu. Your block diagram should resemble Figure 5.

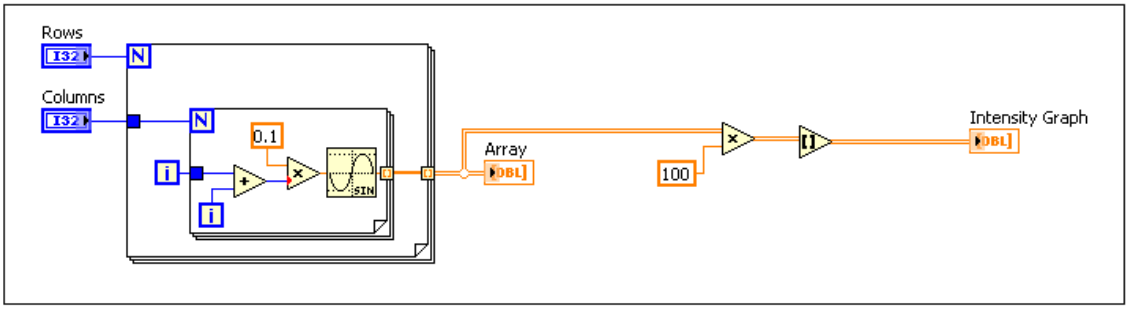

Figure 5. Array Investigation Part 2 VI Block Diagram

6. Save the VI.

7. Switch to the front panel.

8. Enter values for Rows and Columns.

9. Run the VI.

Notice that the VI behaves the same as the solution for Part 1. This is because mathematical functions are polymorphic. For example, because the x input of the Multiply function is a two-dimensional array, and the y input is a scalar, the Multiply function multiplies each element in the array by the scalar, and outputs an array of the same dimension as the x input.

End of Exercise

Notes