Chapter 1. Array and Clusters: Arrays

- An array is a variable-sized collection of data element that are all the same data type.
- A cluster is fixed-size collection of data elements of mixed types.

Array can have one or more dimensions.

- One dimension array has one column. For an n-element one-dimension array, the index is in the range of 0 to n-1.
- **Array Functions** can be fund at Function>Programming palette.
- **Array Control** can be fund at Control>> Modern palette. You can change it as an indicator.

1.1 Creating Array Controls and Indicators

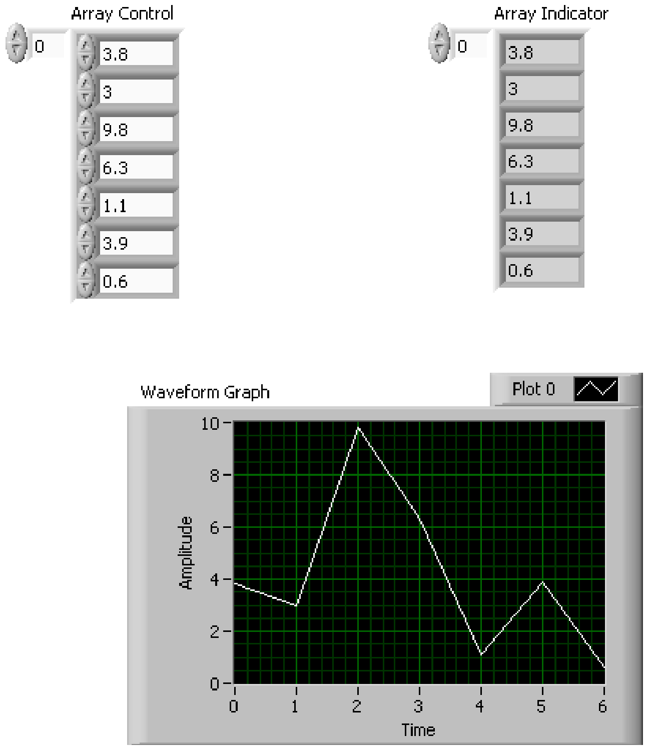
Two steps to create an array control or indicator:

- Select an empty array shell from Controls>Modern palette.
- Define the array: drag a valid object (Such as numeric, Boolean, string) into the array shell.

You can resize the array to display more elements.

Assignment 1:

Create a 7-element array control and indicator as shown in the following figure. The data is input from the array indicate, and display in the array indicator. Also using a wavefromt graph to shown the data of the array. Don’t forget to define the control and indicator arrays.



Array Control

0

3.8
3
9.8
6.3
1.1
3.9
0.6

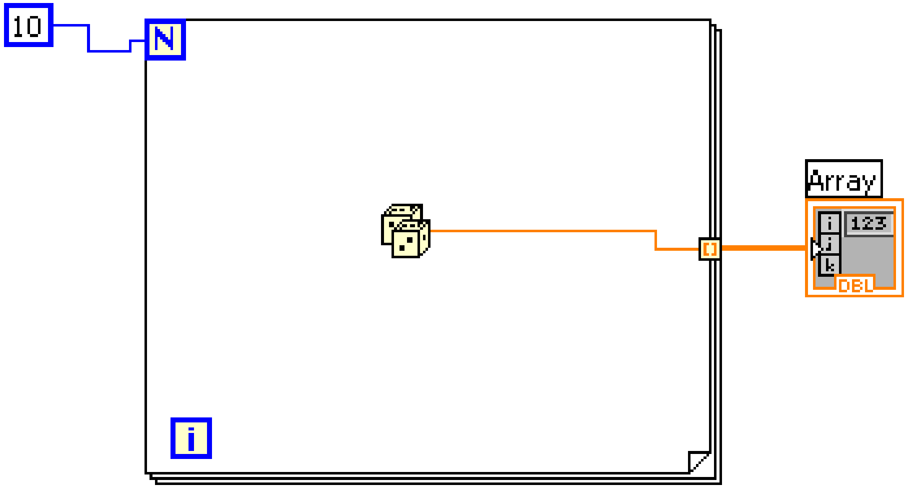
Array Indicator

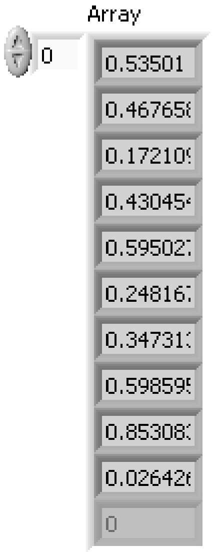
0

3.8
3
9.8
6.3
1.1
3.9
0.6

1.2 Creating Arrays with Loops

You can use a “for loop” to create an array. The following show a 10-element array created by a for loop.

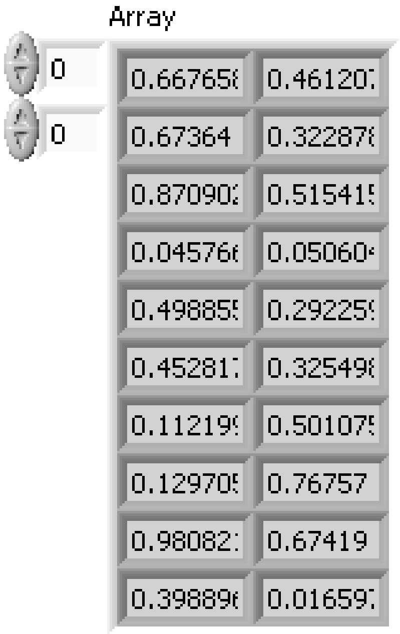




```
0.53501
0.46765
0.17210
0.43045
0.59502
0.24816
0.34731
0.59859
0.85308
0.02642
0
```

Assignment 2

Create a 2-dimensional array with 10x2 element as shown in the following figure. You need using 2 “For loops” and a Random function to create these data. Using a 2-dimentional array indicator to shown the result.   Be noted that the array created must have two columns and 10 rows.



Array

0

0

0.667658	0.461207
0.67364	0.322878
0.870902	0.515415
0.045766	0.050604
0.498855	0.292259
0.452817	0.325498
0.112199	0.501075
0.129705	0.76757
0.980821	0.67419
0.398896	0.016597

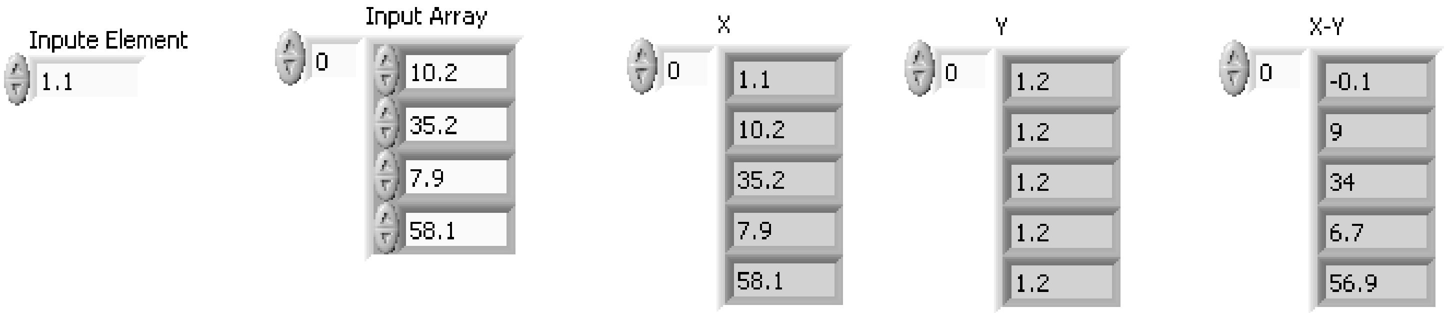
1.3 Array Functions

There are many built-in function can be used to handle arrays. They are located at Functions>programming>Array.

- Array size: This function return the number of elements in the input array.
- Initialize Array: This function create an n-dimensional array with the value you specify, and all elements have the same value.
- Build Array: This function combines multiple arrays or add extra elements to an array. Scalars and arrays can be the inputs.
- Index Array: This function access an element of an array.

Assignment 3

Writing the following VI and running the code. Understand the role of each function, and how the VI works (i.e. how the output data X, Y, and X-Y are generated).



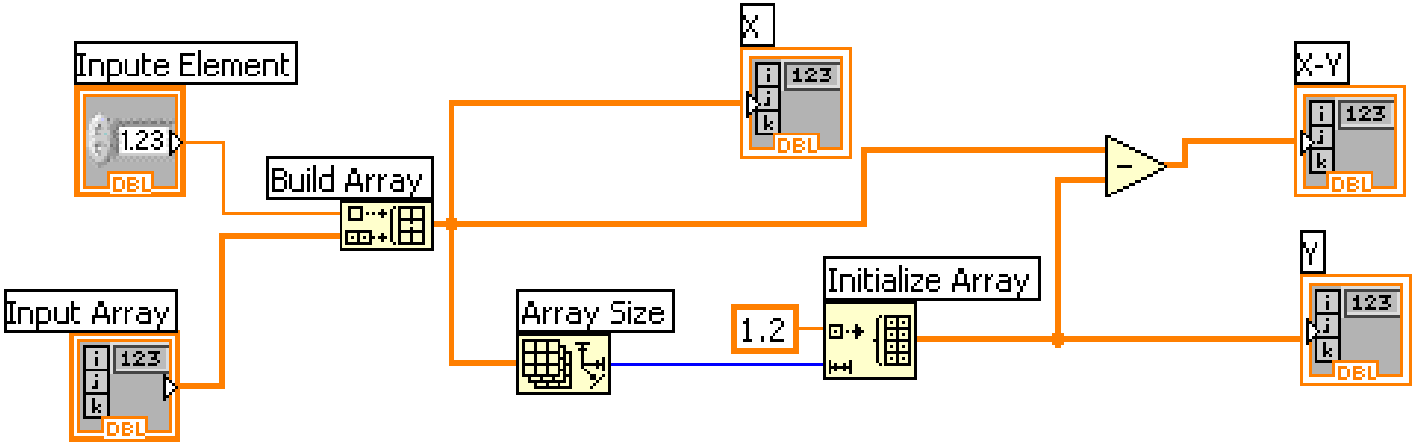
Inpute Element
1.1

Input Array
0
10.2
35.2
7.9
58.1

X
0
1.1
10.2
35.2
7.9
58.1

Y
0
1.2
1.2
1.2
1.2
1.2

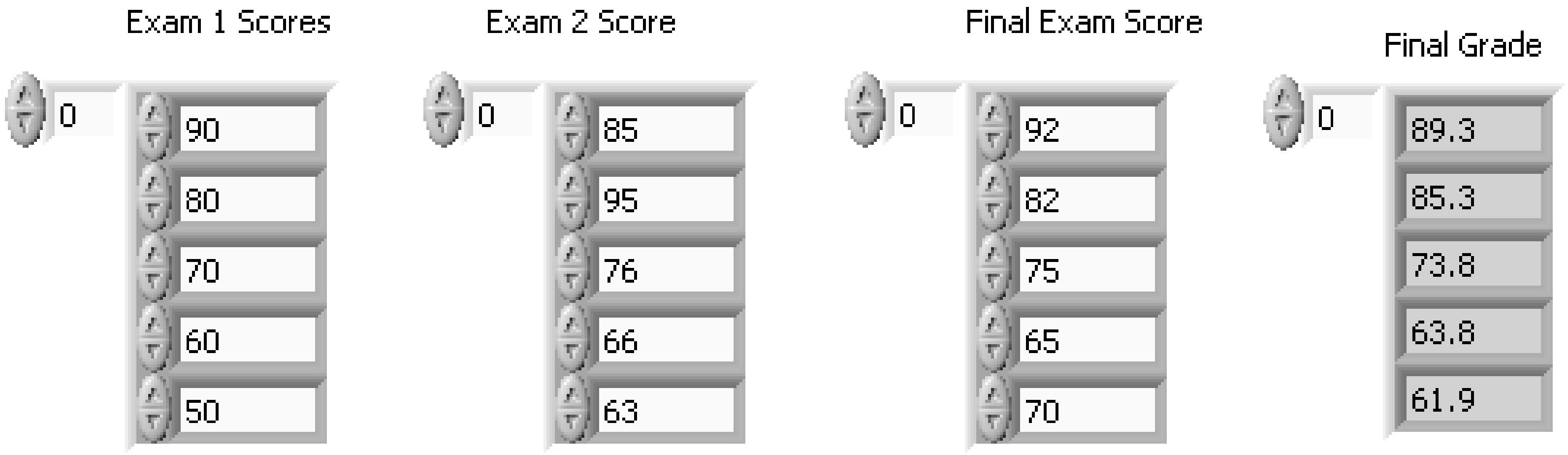
X-Y
0
-0.1
9
34
6.7
56.9



Assignment 4

Writing the a VI that computer the final grades in a course using three input arrays containing students scores on three different exams, using the data as shown in the following front panel.

The input consists of three 1D arrays: Exam 1 Scores (weighted 30% of the final grade); Exam 2 Scores (weighted 30% of the final grade); and Final Exam Scores (weighted 40% of the final grade). The VI should output a 1D array containing the final course grade for each student.

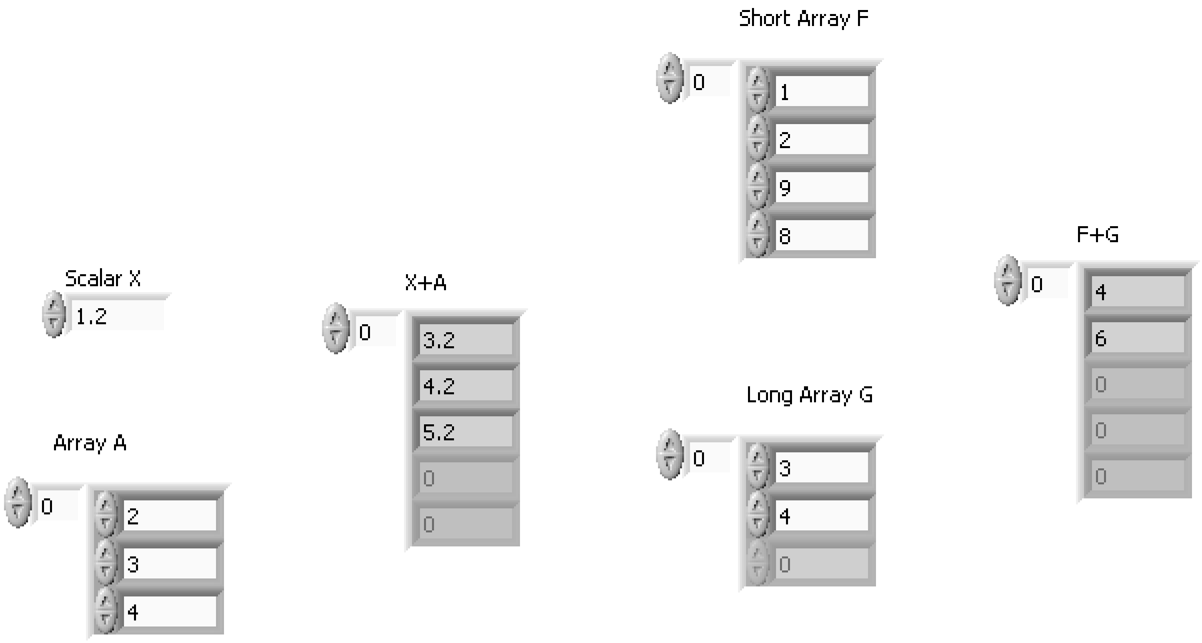


Exam 1 Scores	Exam 2 Score	Final Exam Score	Final Grade
0	0	0	0
90	85	92	89.3
80	95	82	85.3
70	76	75	73.8
60	66	65	63.8
50	63	70	61.9

1.4 Polymorphism

Polymorphism is the ability of certain LabVIEW functions, such as Add, Multiply, and Divide, to accept inputs of different dimensions (or representation). For example, you can add a scalar to an array or add two arrays of different lengths.

The following figure shows some polymorphic combination of the Add function.



Short Array F

0

1

2

9

8

F+G

0

4

6

0

0

0

Scalar X

1.2

X+A

0

3.2

4.2

5.2

0

0

Long Array G

0

3

4

0

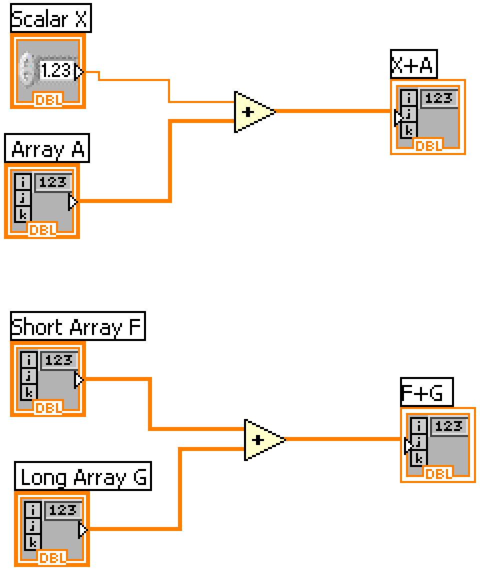
Array A

0

2

3

4



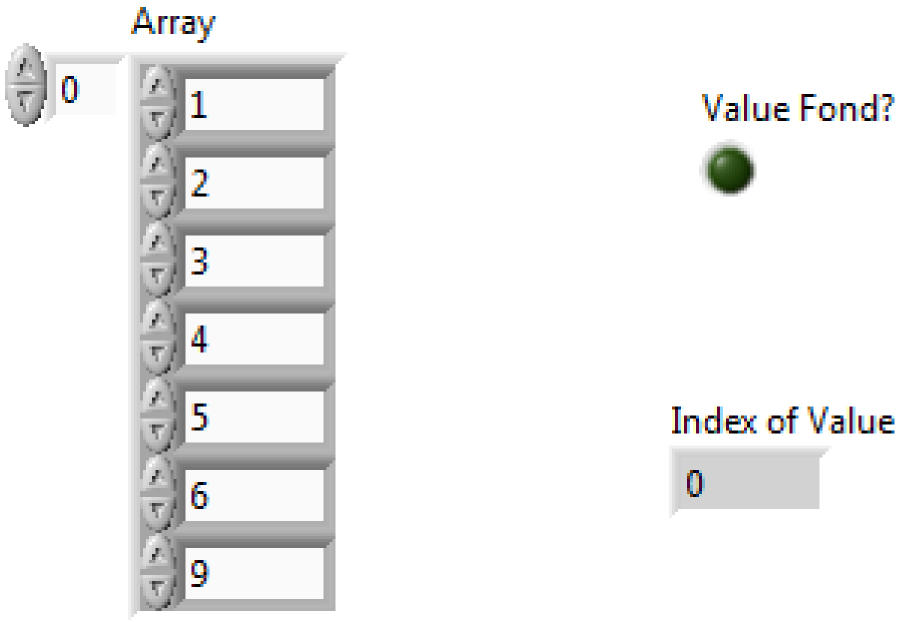
- When two input arrays have different lengths, the output array (after addition, subtraction) will have the same size with smaller of the two input arrays.

Assignment 5

Writing a VI to read an image file of grey-scale intensity and show the image on the front panel. Then, use an array to read all the pixel values (i.e. intensity) of the image. Also use indicators to show the intensity value at the pixel position of (10,10) only and the image size in m and n (i.e. the array size). Also find the pixel positions of the maximum and minimum intensities, respectively. To convert an image to array, you may need to use the function “IMAQ ImageTo Array”

Assignment 6 (an input value is needed for the search)

Construct a VI that searches the values of an input array of numbers for a specified value. If the specific value is found in the array, the VI should indicate that by turning on an LED and by indicating the corresponding index of the array. Using the following figure as a guide. Write this code that include 2 approaches in this code: 1)Using the standard function “Search 1D Array”; 2) Using a “for loop” to find the index.



Array

0

1
2
3
4
5
6
9

Value Fond?

Index of Value

0

Assignment 7 (Extra assignment)

Create a “universal VI” that calculate the cross product of two 3-dimensional vectors.

Hint: if the two vectors are denoted by v_1 and v_2, where

$$v_1 = \begin{bmatrix} v_1(0) \\ v_1(1) \\ v_1(2) \end{bmatrix} \text{ and } v_2 = \begin{bmatrix} v_2(0) \\ v_2(1) \\ v_2(2) \end{bmatrix},$$

Then the cross product is given by

$$v_1 \times v_2 = \begin{bmatrix} v_1(1)v_2(2) - v_1(2)v_2(1) \\ v_1(2)v_2(0) - v_1(0)v_2(2) \\ v_1(0)v_2(1) - v_1(1)v_2(0) \end{bmatrix}$$

Assignment 8 (Extra assignment)

Construct a “universal VI” that performs matrix multiplication for two input Matrices **A** and **B**. The Matrix A is n x m matrix, and the matrix **B** is an m x p matrix. The resulting matrix C is an n x p matrix, where **C=AB**. Double check your math by comparing your calculation using arrays with FOR loops and math functions with the results of AxB.vi found in the “**Mathematics>>Linear Algebra**” palette.