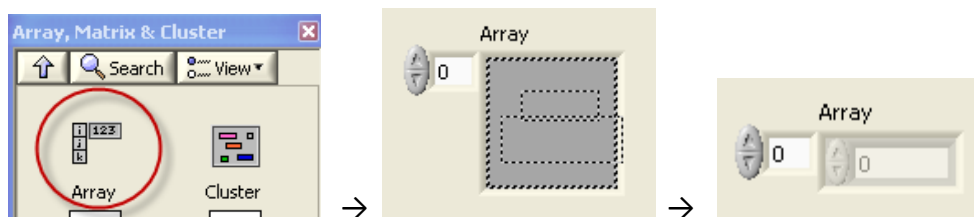


# LabVIEW Exercises

## LabVIEW Arrays

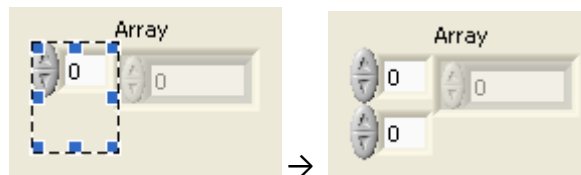
Working and manipulating with Arrays is an important part in LabVIEW development. Arrays are very powerful to use in LabVIEW. In all your applications you would probably use both One-Dimensional Arrays and Two-Dimensional Arrays.

On the Front Panel using the Control palette we can create an array as follows (Array, Matrix & Cluster subpalette):

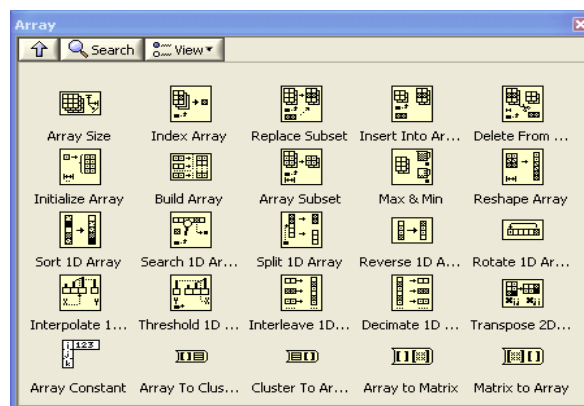


You drag and drop the empty Array on the Front Panel, next you find a Control or Indicator (Numeric, String, Boolean, etc,) and drag it into the empty Array. You can create an Array of (almost) any kind of Control or Indicator.









2D or multidimensional Array? Just drag the mouse in the Index display to the left and increase the dimension.



On the Block Diagram we have the following Array palette available from the Functions palette in LabVIEW:

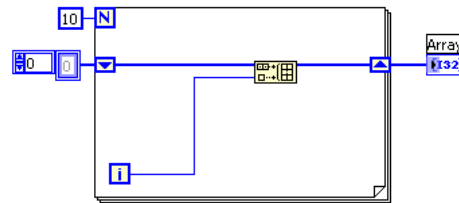
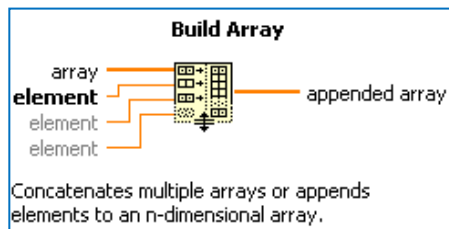


Use the Array functions to create and manipulate arrays. The most useful Array functions are:

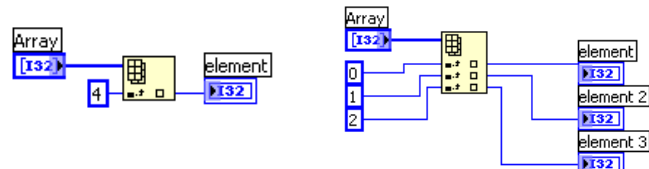
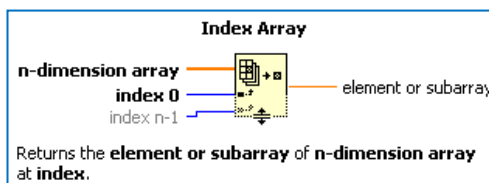
 Array Size	 Index Array	 Delete from Array	 Search 1D Array
 Initialize Array	 Build Array	 Array Subset	 Array Constant

All these functions are basic (but very useful) array functions you will probably be using in all your applications and VIs.

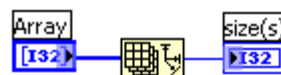
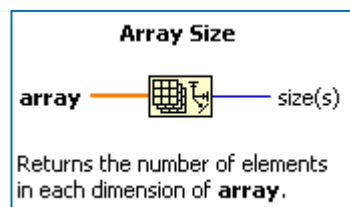
**Task: Build Array.** This function concatenates multiple arrays or appends elements to an n-dimensional array. Try the simple example below. This example using the Build Array function inside a For loop in order build an array with 10 elements.



**Task: Index Array.** This function returns the element or subarray of n-dimension array at index. It is always useful to find a specific value in an array. The Index Array is extendible, so you can drag it out to find more than one elements. Try the simple example below.



**Task: Array Size.** This function returns the number of elements in each dimension of array. Try the simple example below. The example finds the size of an arbitrary array.



**Task Auto-indexing.** If you wire an array to a For Loop, you can read and process every element in that array by enabling auto-indexing. You also can enable auto-indexing by

configuring a For Loop to return an array of every value generated by the loop. Create a simple example in order to see the difference.

