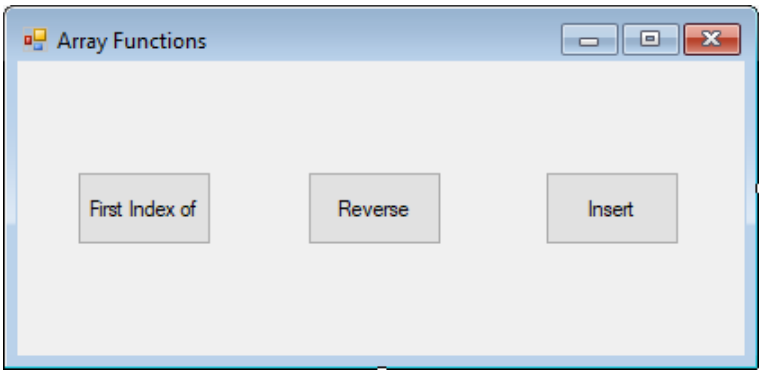
**CIS 366 Introduction to .NET Development using C# (Spring 2019)**

**Assignment 7**

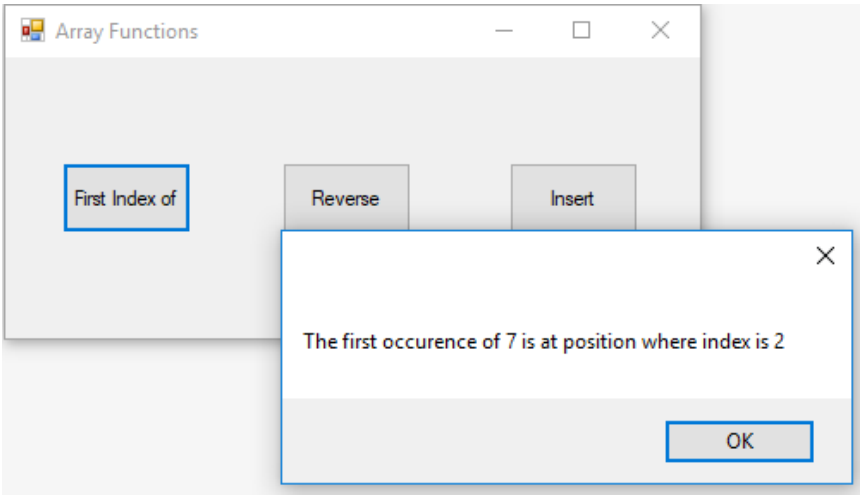
## Requirement

This assignment is to design a windows application to design some array functions. You will design them and call them in the main function to test them. In this assignment, you will use Visual Studio 2015 and write C# code for following functionalities:

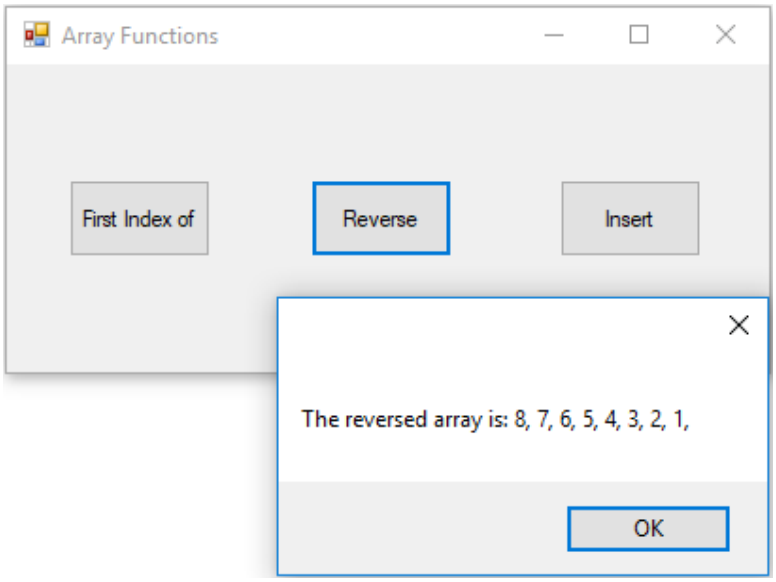
1. Build a user interface that looks like this. It allows users to click on buttons to perform certain array function.



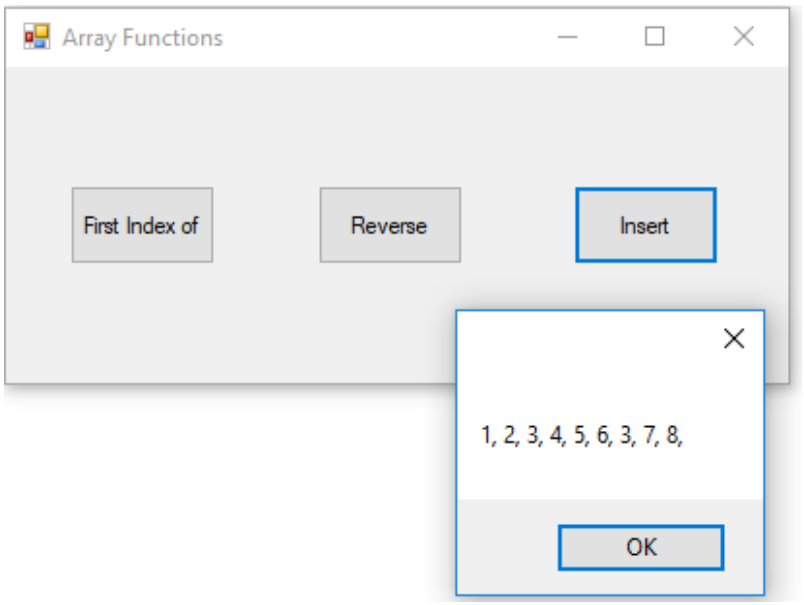

2. "First Index of" function takes two inputs: (1) an integer array, and (2) an integer element. This function is to find the first occurrence of the input element in the input array and return the index of that occurrence. If the element does not appear in the array, return -1. In the button click function of "First Index of", define a testing array (e.g. {5, 4, 7, 2, 6, 9, 7, 3}) and an element to find the index for, say, 7. Call this function and show a message like this in the message box.



3. “Reverse” function takes an integer array as the input. This function is to store elements in the input array in a reversed order. For example, an array of {1, 3, 2, 4, 5} will be converted to {5, 4, 2, 3, 1}. Hint: You can pass the input array **by reference** and reverse the elements inside it and return the same array reference for the reversed array. On the algorithm of reversing array elements, you can do the following: have a loop from index 0 to array.Length / 2, in each iteration, swap the element at index [i] and the element at index [array.Length - i – 1]. In the button click function of “Reverse”, define a testing array (e.g. {1, 2, 3, 4, 5, 6, 7, 8}) and call this function and show a message like this in the message box.




4. “Insert” function takes three inputs: (1) an integer array, (2) an integer element to insert, and (3) the index where the element to be inserted at. This function is to insert the element into the input array at the specified index. Hint: You can create a new array whose length equals to the length of the input array plus one. Then copy all elements before the specified insert index, insert the new element, and copy the rest of the array elements to the new array, and finally return the new array reference. In the button click function of “Index”, define a testing array (e.g. {1, 2, 3, 4, 5, 6, 7, 8}), then call this function to insert an element 3 at the index 6. Then show a message containing the elements in the new array like this in the message box.




5. Although not required, you might find it useful to prepare several other functions such as *printArray* and *swap*. You can copy them from our class exercises.

**Submission**

Zip your ENTIRE project folder and name your zipped file to (yourlastname)_a7. Submit your zipped file to the Blackboard dropbox as an attachment.