LAB GOALS

More advanced operations on arrays.

1) Search
2) Sort
3) Passing array parameters to procedures

Step 1: Create a new console project named “Console_Application_Array_Manipulation_Advanced”. Make sure to save your new project in your I101/B100 folder.

Step 2: Start by copying and pasting the following code in your console application (this was developed in the previous lab).

```vbnet
Option Explicit On
Option Strict On
Module Module1
    Sub Main()
        Dim TestScores(10) As Integer
        Console.WriteLine("Array length = {0}", TestScores.Length)
        Console.WriteLine("Array upper bound = {0}", TestScores.GetUpperBound(0))
        Initialize_Array(TestScores)
        Print_Array(TestScores)
        Randomize_Array(TestScores)
        Print_Array(TestScores)
        Console.ReadLine() ' pause the program
    End Sub
End Module
```

```
Private Sub Initialize_Array(ByRef TheArray() As Integer)
    For Index = 0 To TheArray.GetUpperBound(0)
        TheArray(Index) = -1
    Next Index
End Sub
```

```
Private Sub Print_Array(ByVal TheArray() As Integer)
    For Index = 0 To TheArray.GetUpperBound(0)
        Console.WriteLine(TheArray(Index))
    Next Index
End Sub
```

```
Private Sub Randomize_Array(ByRef TheArray() As Integer)
    Randomize() 'Generate a Random Seed
    For Index = 0 To TheArray.GetUpperBound(0)
        TheArray(Index) = CInt(Rnd() * 100)
    Next Index
End Sub
```

Run: Compile and Run your program to make sure it does not have any syntax errors.

Step 3: After reviewing the above code to make sure you understand how each of the above procedures work, proceed to
make the following additional operations for the above array of TestScores. For each of the operations below, take about 3 minutes to brainstorm the algorithm with one or more member of your team. Write the algorithm (do not code yet), and ask any questions that arise. After your questions have been answered proceed to code individually, asking for help from me or your team members if necessary.

1) Write a procedure named Read_Scores() which asks the user to manually enter the test score for each student. Call the Read_Scores() procedure from the main program, and then call the Print_Array() to verify the test scores entered by the user.

2) Write a function named Min_Score() which finds and returns the minimum score in the array.

3) Write a function named Max_Score() which finds and returns the maximum score in the array.

4) Write a function named Average_Score() which calculates and returns the average score in the array.

5) Write a procedure named Search_For_Score(ATestScore) which searches the entire array for the particular test score. The procedure should print the array location (index) in which the test score is found.

Note: Compile and Run and Test your program after every operation is coded.

Step 4: We will write the code for this next operation as a class.

6) Write a procedure named Sort_Scores() which will sort the array elements in ascending order.