

LAB GOALS

Array Manipulation

Open up Microsoft Visual Basic .NET.

Step 1: Create a new Console project named “ArrayManipulation”. Don’t forget..

Option Explicit On

Option Strict On

Step 2: Working With Arrays:

Start by copying and pasting the program below into your project. Compile and run it. Make sure you understand how it works. Try changing the loop upper bound. For example try using 5. Now try using 15. Do you notice any problems?

Module Module1

```
Sub Main()
    Dim Index As Integer
    Dim TestScores(10) As Double
    'This declaration makes available 10 storage
    'locations (Actually 11) in which to put floats.

    'looking at array properties
    Console.WriteLine("Array length = {0}", TestScores.Length)
    Console.WriteLine("Array upper bound = {0}", TestScores.GetUpperBound(0))

    'The individual array members or cells of the array defined above are denoted by
    'TestScores(1), TestScores(2), TestScore(3), . . . TestScore(10).

    'The index values 1, 2, .. are used to access the content of an array cells are also called subscripts. The type of the
    'data that is actually stored in the array is called the base type. In the array above, the base type is Double,
    'but it can be any type, i.e. a programmer can set up an array to hold any VB data type. All members of a
    'single array must be the same type.

    'One of the features of arrays that makes them so powerful is that variables can be used to specify an array location.
    'This allows a loop to very efficiently process the elements of the array.

    'For example the loop below can be used to initialize all the elements of the above array to 0.0

    For Index = 0 To 10
        TestScores(Index) = 0.0
    Next Index

    'The loop below will print the values in the array. Note the use of the Index loop variable in
    'extracting the array elements.
    For Index = 0 To 10
        Console.WriteLine(TestScores(Index))
    Next Index

    'The loop below will Read some new values from the keyboard and store them into the array locations
    For Index = 0 To 10
        Console.Write("Enter array element ")
        Console.Write(Index)
        Console.Write(": ")
        TestScores(Index) = Console.ReadLine()
    Next Index

    'Now print the values again
    For Index = 0 To 10
        Console.WriteLine(TestScores(Index))
    Next Index

    'The loop blow will Search for a given value in an array
    Dim value As Double
    Console.Write("Enter the value to search for:")
    value = Console.ReadLine() 'value to search for

    For Index = 0 To 10
        If TestScores(Index) = value Then
            Console.WriteLine("location {0}, Found it!", Index)
        Else
            Console.WriteLine("Location {0}, Not Found..", Index)
        End If
    Next Index
End Sub
```

```

        End If
    Next Index

    Console.ReadLine()
End Sub

End Module

```

Step 3: Creating the DRIVER program:

Create a main program with the following procedure calls. Note that all of the procedure calls are currently commented.

```

Sub Main()
    Dim Index As Integer
    Dim TestScores(10) As Double

    InitializeArray(TestScores)
    PrintArray(TestScores)

    LoadArray(TestScores)
    PrintArray(TestScores)

    SearchArray(TestScores)

    Console.ReadLine()
End Sub

```

Step 4: Creating A Series of STUB Procedures:

Create a STUB procedure for each of the above procedure calls. Each stub should have two writeline statements in them such as:

```

Console.WriteLine("Entering: -----> Procedure_Name")
'Your code goes here...
Console.WriteLine("Exiting: -----> Procedure_Name")

```

Step 5: Programming the STUB Procedures:

Now that we have our STUB procedure created, we are ready to program and test each one. Take each of the code segments from the main program and transfer it to its proper sub procedure. For example the code that initializes the array should be removed from the main program and placed in the Initialize() sub procedure.

```

Private Sub InitializeArray(ByRef TestScores() As Double)
    Dim Index As Integer

    For Index = 0 To 10
        TestScores(Index) = 0.0
    Next Index
End Sub

```

Then compile and run your program. Make sure to “UN-COMMENT” the procedure call before compiling and running your program!

Complete the rest of the sub procedures. Pay attention to whether you should pass the array by value or by reference.