LAB GOALS

Understanding Loops (Do While loop and For-Next loop)
Reviewing Conditionals and Operators

Problem 1)

Using a WHILE loop; write a console application which will display all the odd numbers between
22 and 97 (inclusive).

This problem can be broken down into two parts. One part deals with figuring out if a number is
odd or even. The second part deals with creating a loop that correctly examines numbers
between 22 and 97 (inclusively).

To accomplish part one, we can use the MOD operator. For example we can check to see if
(number MOD 2) = 0, if so, the number is even.

To accomplish part two, we need to create a WHILE loop that correctly steps through the
numbers 22 to 97.

Let’s start by working on the loop first. Let’s write a loop that simply displays all the numbers
between 22 and 97.

```vba
Option Explicit On
Option Strict On
Module Module1
  Sub Main()
    Dim number As Integer
    number = 1
    While (number < 100)
      If (number >= 22 And number <= 97) Then
        If ((number Mod 2) = 1) Then
          Console.WriteLine(number)
        End If
      End If
      number = number + 1
    End While
    Console.ReadLine()
  End Sub
End Module
```

Note, the above algorithm is not that efficient since it goes though all numbers from 1 to 100, and
checks to see if the number is within 22 and 97.

Let’s improve the above algorithm a little. The highlighted code below (in YELLOW) can replace
the highlighted code above (in GREEN) and produce a more efficient algorithm:

```vba
number = 22
While (number <= 97)
  If ((number Mod 2) = 1) Then
    Console.WriteLine(number)
  End If
  number = number + 1
End While
```

**NOTE:** The efficiency comes from having a smaller loop (number = 22, While (number <= 97)) as
well as reducing the number of comparisons (If (number >= 22 And number <= 97) Then).
Problem 2)

Using a FOR loop; write a console application which will display the ASCII code value of the letters (A through Z).

```vbnet
Option Explicit On
Option Strict On
Module Module1
    Sub Main()
        Dim ASCII_Value As Integer
        For ASCII_Value = Asc("A") To Asc("Z")
            Console.WriteLine(ASCII_Value)
        Next ASCII_Value
        Console.ReadLine()
    End Sub
End Module
```

Problem 3)

Write a program which asks the user to enter a positive integer and it prints out if the number is a prime or not.

```vbnet
Option Explicit On
Option Strict On
Module Module1
    Sub Main()
        Dim number As Integer
        Dim PRIME As Boolean = True
        Console.Write("Please enter a number: ")
        number = CInt(Console.ReadLine())
        If number = 1 Or number = 2 Or number = 3 Then
            PRIME = True
        Else
            For index = (number - 1) To 2 Step -1
                If number Mod index = 0 Then ' if the number is divisible by a factor
                    PRIME = False
                    Exit For
                End If
            Next
        End If
        If (PRIME = True) Then
            Console.WriteLine("{0} is a prime", number)
        Else
            Console.WriteLine("{0} is NOT a prime", number)
        End If
        Console.ReadLine()
    End Sub
End Module
```

Test the above algorithm by running the program several times with both prime and non-prime numbers.
Team Exercises

Problem 4
Step 1) Together with your team members write a pseudocode algorithm which includes a **WHILE** loop that inspects all the integer numbers between 0 and 100 and prints out all the number that are divisible by (2 or 3 or 4).

Step 2) Together with your team members, translate the pseudocode to Visual Basic. Remember to compile and test your program often!

Step 3) Individually, Write a **WHILE** loop that inspects all the numbers between 0 and 100 and prints out all the number that are divisible by (2 and 3 and 4).

Problem 5
Step 1) Together with your team members write a pseudocode algorithm which includes a **FOR** loop that inspects all the numbers between 0 and 100 and prints out all the number that are divisible by (2 or 3 or 4).

Step 2) Together with your team members, translate the pseudocode to Visual Basic. Remember to compile and test your program often!

Step 3) Individually, Write a **FOR** loop that inspects all the numbers between 0 and 100 and prints out all the number that are divisible by (2 and 3 and 4).