

PROGRAMMING TEST 3 MS

ALL SOLUTIONS WERE CRAFTED AND CODED BY
YOURS FAITHFULLY

PROGRAMMING IS AN ART WHICH CANNOT BE REPLICATED

- Eng Tino Chibi

Question 1: Functions

Write a user-defined function in [VB.NET](#) that takes two integers as input and returns their sum.

```
Imports System.Console
0 references
Module Module1
    1 reference
    Function sumfunction(ByVal num1 As Integer, ByVal num2 As Integer) As Integer
        Dim result As Integer
        result = num1 + num2
        Return result
    End Function
    0 references
    Sub Main()
        Dim firstnum, secondnum, sum As Double
        WriteLine("enter the first number")
        firstnum = ReadLine()
        WriteLine("enter the second number")
        secondnum = ReadLine()
        sum = sumfunction(firstnum, secondnum)
        WriteLine("the sum is " & sum)
        ReadKey()
    End Sub
End Module
```

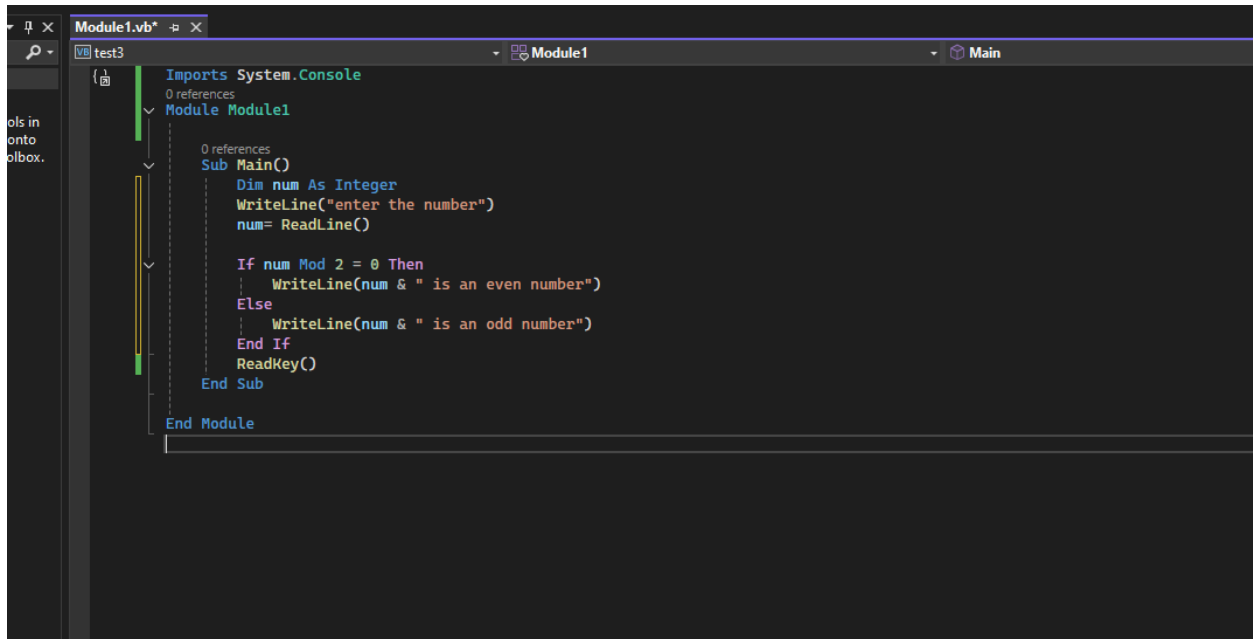
Question 2: Loops

Write a [VB.NET](#) code snippet that uses a For loop to print the numbers 1 to 10.

```
Module1.vb  test3  Module1  Main
Imports System.Console
0 references
Module Module1
    0 references
    Sub Main()
        For k As Integer = 1 To 10
            WriteLine(k)
        Next
        ReadKey()
    End Sub
End Module
```

Question 3: If Statement

Write a [VB.NET](#) code snippet that uses an if statement to check if a given number is even or odd.



The screenshot shows a Visual Studio IDE with a file named 'Module1.vb'. The code is as follows:

```
Imports System.Console

Module Module1

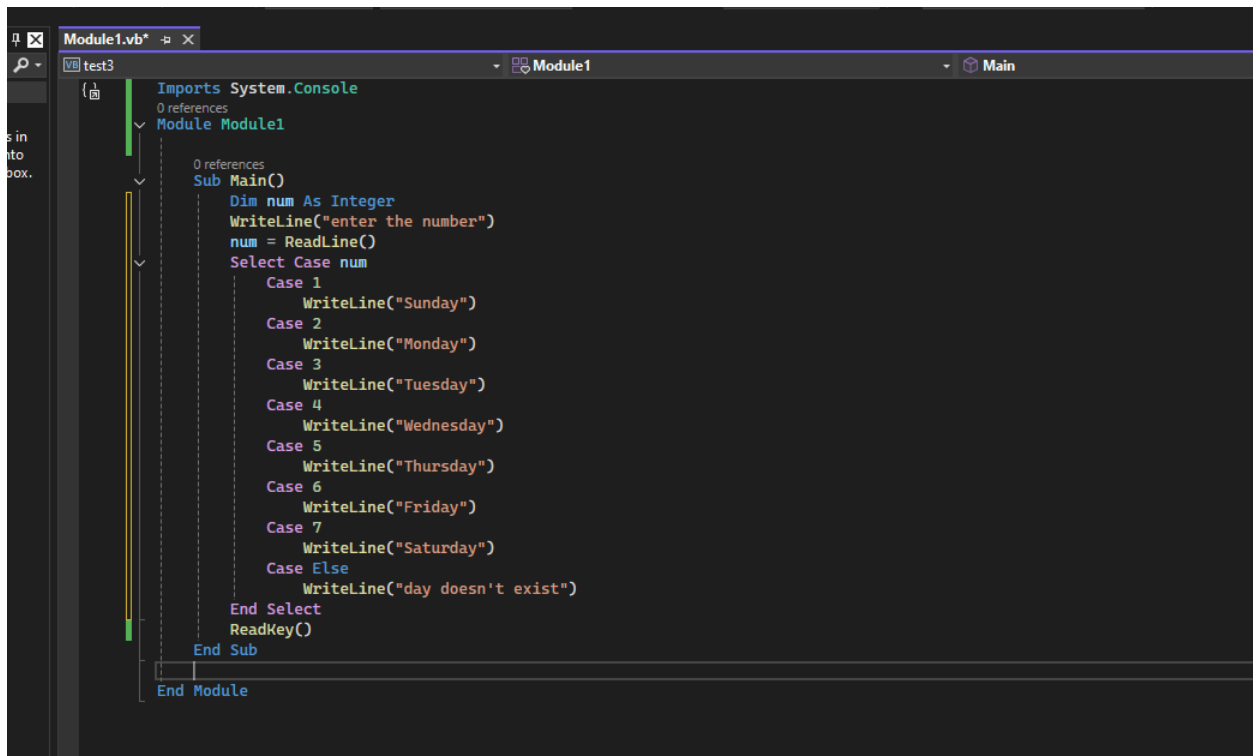
    Sub Main()
        Dim num As Integer
        WriteLine("enter the number")
        num = ReadLine()

        If num Mod 2 = 0 Then
            WriteLine(num & " is an even number")
        Else
            WriteLine(num & " is an odd number")
        End If
        ReadKey()
    End Sub

End Module
```

Question 4: Select Case

Write a [VB.NET](#) code snippet that uses a select case statement to determine the day of the week based on a given integer value (1-7).



The screenshot shows a Visual Studio IDE with a file named 'Module1.vb'. The code is as follows:

```
Imports System.Console

Module Module1

    Sub Main()
        Dim num As Integer
        WriteLine("enter the number")
        num = ReadLine()
        Select Case num
            Case 1
                WriteLine("Sunday")
            Case 2
                WriteLine("Monday")
            Case 3
                WriteLine("Tuesday")
            Case 4
                WriteLine("Wednesday")
            Case 5
                WriteLine("Thursday")
            Case 6
                WriteLine("Friday")
            Case 7
                WriteLine("Saturday")
            Case Else
                WriteLine("day doesn't exist")
        End Select
        ReadKey()
    End Sub

End Module
```

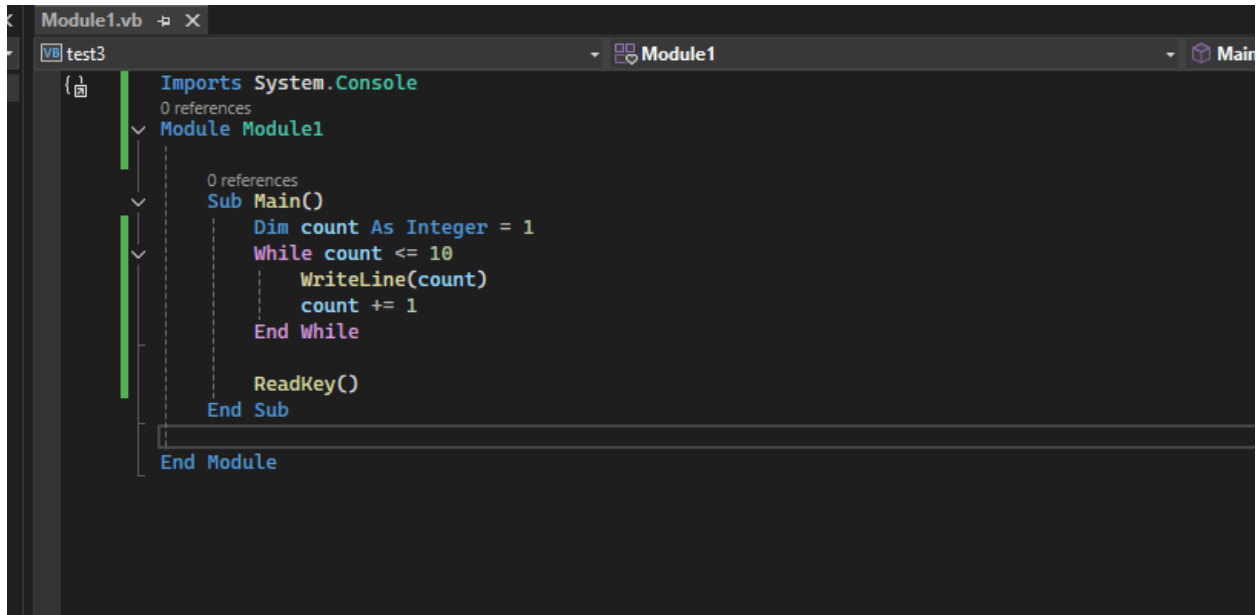
Question 5: Inbuilt Functions

What is the purpose of the `Trim()` function in [VB.NET](#)?

-----iyi pindurawo wega

Question 6: Loops

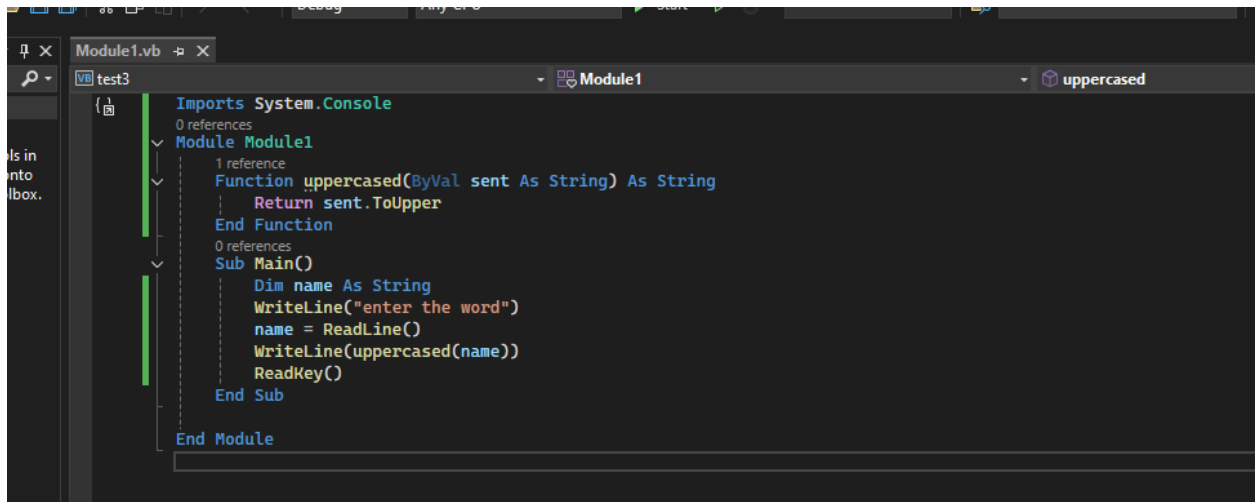
Write a [VB.NET](#) code snippet that uses a While loop to print the numbers 1 to 10.



```
Module1.vb
VB test3
Imports System.Console
0 references
Module Module1
    0 references
    Sub Main()
        Dim count As Integer = 1
        While count <= 10
            WriteLine(count)
            count += 1
        End While
        ReadKey()
    End Sub
End Module
```

Question 7: Functions

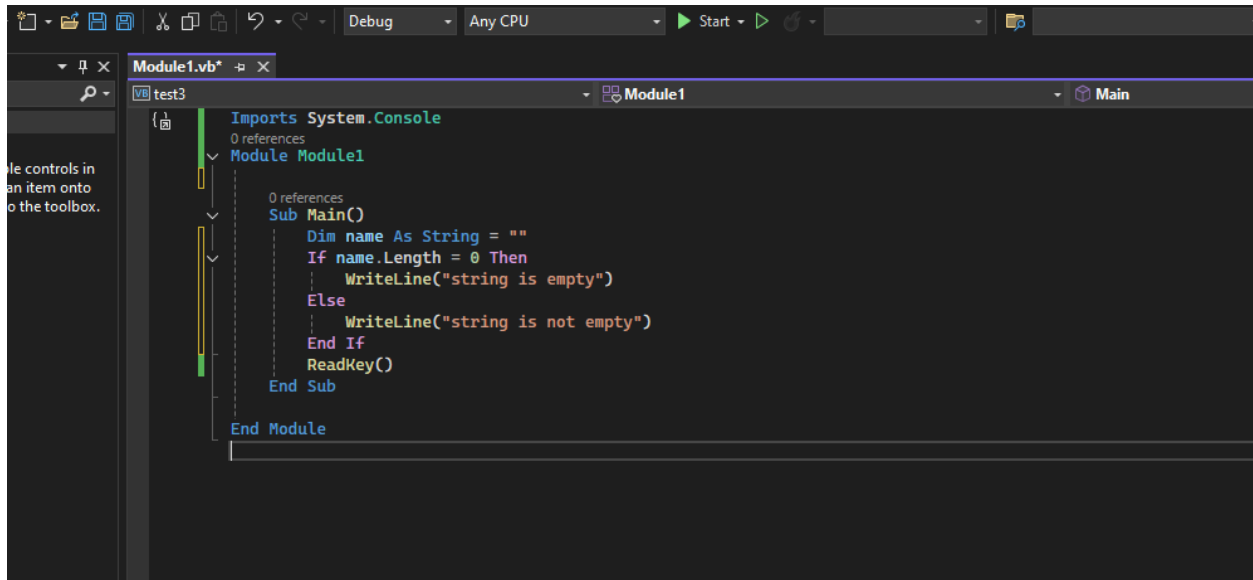
Write a user-defined function in [VB.NET](#) that takes a string as input and returns the string in uppercase.



```
Module1.vb
VB test3
Imports System.Console
0 references
Module Module1
    1 reference
    Function uppercased(ByVal sent As String) As String
        Return sent.ToUpper
    End Function
    0 references
    Sub Main()
        Dim name As String
        WriteLine("enter the word")
        name = ReadLine()
        WriteLine(uppercased(name))
        ReadKey()
    End Sub
End Module
```

Question 8: If Statement

Write a [VB.NET](#) code snippet that uses an if statement to check if a given string is empty or not.



The screenshot shows the Visual Studio IDE with a VB.NET project. The code is written in a dark-themed editor. The code snippet is as follows:

```
Imports System.Console
0 references
Module Module1
    0 references
    Sub Main()
        Dim name As String = ""
        If name.Length = 0 Then
            WriteLine("string is empty")
        Else
            WriteLine("string is not empty")
        End If
        ReadKey()
    End Sub
End Module
```

The code is enclosed in a module named `Module1` with a sub `Main()`. It declares a string variable `name` and uses an `If` statement to check if its length is 0. If true, it writes "string is empty" to the console; otherwise, it writes "string is not empty". The `ReadKey()` method is called at the end of the `Main` sub to keep the console window open.

Question 9: Select Case

Write a [VB.NET](#) code snippet that uses a select case statement to determine the grade of a student based on their score (A: [90-100](#), B: [80-89](#), C: [70-79](#), D: [60-69](#), F: below 60).

```
VB test3
Imports System.Console
0 references
Module Module1
1 reference
    Function grade(ByVal num As Integer) As String
        Select Case num
            Case 90 To 100
                Return "A"
            Case 80 To 89
                Return "B"
            Case 70 To 79
                Return "C"
            Case 60 To 69
                Return "D"
            Case 0 To 59
                Return "FAIL"
            Case Else
                Return "INVALID"
        End Select
    End Function
End Function
0 references
Sub Main()
    Dim mark As Integer
    WriteLine("enter your mark")
    mark = ReadLine()
    WriteLine(grade(mark))
    ReadKey()
End Sub
End Module
```

Question 10: Loops

Write a [VB.NET](#) code snippet that uses a Do-While loop to print the numbers 1 to 10.

```
Module1.vb*
VB test3
Imports System.Console
0 references
Module Module1
0 references
    Sub Main()
        Dim count As Integer = 1
        Do While count <= 5
            WriteLine(count)
            count += 1
        Loop
        ReadKey()
    End Sub
End Module
```